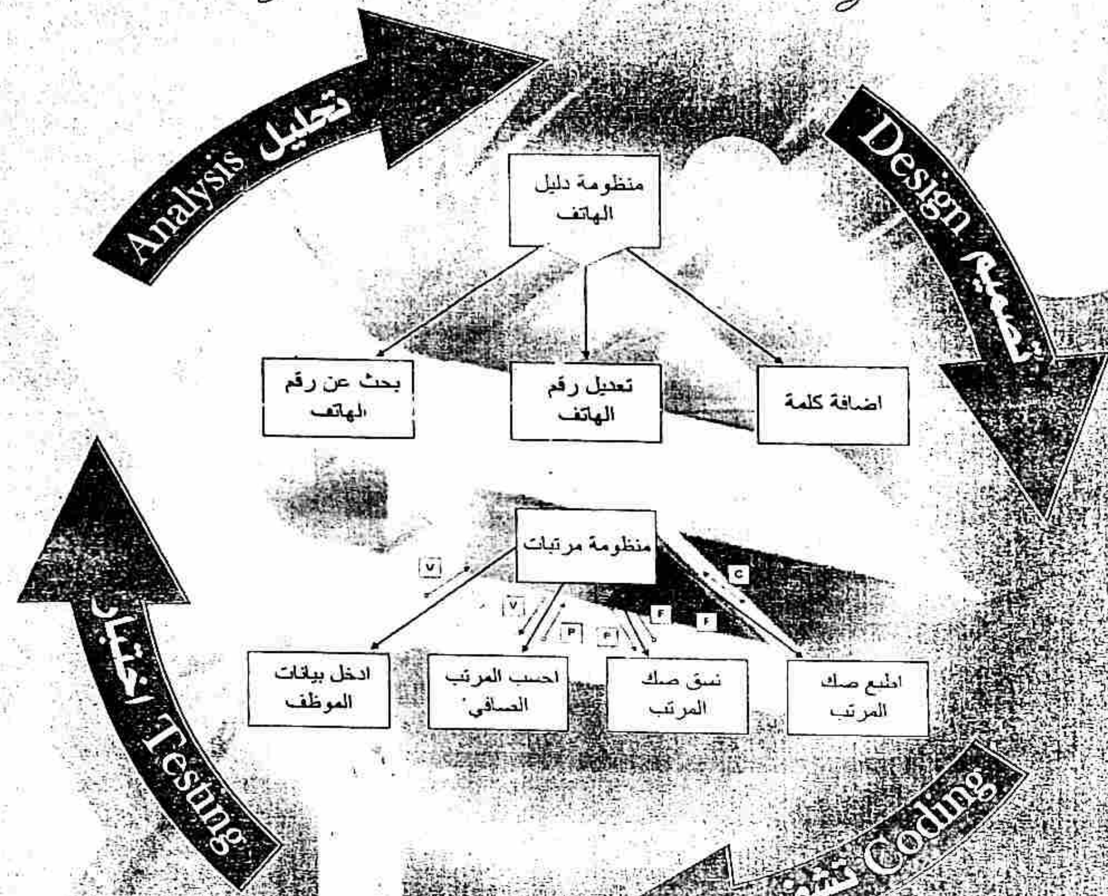
تحليل النظم

systems analysis



الطبعة الثانية
2011

تأليف: أ. إمحمد بن زاهية
Mehemed ben zahid

مراجعة أ.د. عمر زرتي
Omar zorty

المحتويات

تمهيد

1-لمحة عن تحليل النظم

2-نظم المعلومات information systems

تمهيد

يعتبر موضوع (تحليل النظم) ذا أهمية قصوى في إعداد المنظومات ، فهو يهتم بدراسة طرق تطوير أداء منظومات الحاسوب كوسيلة يمكن الاعتماد عليها بجدارة في أداء الأعمال المتعددة.

إن طريقة تخطيط وإعداد المنظومات يمكن أن تصبح مملة ومتعبة نظرا لتعقد التطبيقات التي تحتاج إلى فريق عمل من معدي النظم , وهذا قد يقود إلى مشاكل تأخر تسليم المنتوج وفشل في تلبية المتطلبات.

لذلك فقد أصبح لزاما وضروريا أن يتم اتباع أسلوب سليم في إعداد البرمجيات.

وأصبح موضوع هذا الكتاب (تحليل النظم) اساسيا في جميع مناهج علم الحاسوب في الجامعات والمعاهد العليا.

وقد أعطت الثورة الجديدة في تقنية المعلومات والانترنت والوسائل المتعددة واتصالات البيانات لموضوع تحليل النظم أهمية أكثر.

لقد حاولنا قدر جهدنا أن يكون هذا الكتاب مبسطا بما يكفي بحيث تصبح مادة الموضوع في حدود المعقول كمقرر في مستوى التدريس الجامعية ونعتقد أن الكتاب سيستفيد منه ليس الطلبة فقط بل أيضا المبرمجون المحترفون ومدراء المشاريع وغيرهم.

لقد أعددنا هذا الكتاب بالعربية ولكن احتفظنا بالمصطلحات الانجليزية ، والسبب في ذلك يرجع إلى أن المصطلحات العربية لازالت غير مستقرة في هذا العلم المتغير ، بينما المصطلحات الإنجليزية هي المرجع عالميا.

كما يتميز الكتاب بمراعاة الناحية العملية. حيث تم عرض الأدوات التي تمت تجربتها فقط وأثبتت فائدتها.

ولقد اعتمد المؤلف على الإنترنت كمصدر أساسي للمعلومات إلى جانب الكتب المرجعية في مجال (تحليل النظم).

هذا الكتاب يساعد الطالب الراغب في مواصلة تعليمه العالي.

وقد اتبعت مواضيع الكتاب التسلسل المنطقي التالي:

- محلل النظم ودوره ونشاطاته التي يجب أن يقوم بها قبل واثناء إعداد المنظومة.
- نظم المعلومات وعلاقتها بالحاسوب خاصة بعد تطور تقنية المعلومات (شبكات، انترنت، ووسائل متعددة).
- دورة حياة إعداد المنظومة (تخطيط، تحليل، تصميم، تنفيذ) مع التأكيد على التوثيق في كل مرحلة.
- تخطيط المشاريع ودوره المهم في إعداد المنظومات (الجدولة، تقدير الوقت، حدود المنظومة، ..).
- أدوات التحليل مثل مخطط انسياب البيانات والانجليزية المركبة وشجرة القرار وجدول القرار وقاموس البيانات.
- نموذج العرض التجريبي كأداة فاعلة في الحصول على متطلبات المستخدم والواجهة المناسبة مع مناقشة مزايا وعيوب هذه الطريقة.
- التحليل الشيئي كطريقة جديدة في تحليل النظم تستغل مفهوم إعادة استعمال البرمجيات في إعداد المنظومات بطريقة اقتصادية

1

لمحة عن تحليل وتصميم النظم

1. مقدمة
2. مراحل إعداد المنظومة
3. نشاطات التحليل
4. نشاطات التصميم
5. أهداف التحليل والتصحيم
6. محلل النظم
7. أدوات تحليل وتصميم النظم
8. إرشادات في التحليل والتصميم
9. إدارة المشروع
10. أسئلة للمراجعة

1.1 مقدمـــة

موضوع هذا الكتاب هو تحليل وتصميم النظم. فما هو النظام؟ وماهو التحليل ؟ وما هو التصميم؟

النظــام System

هو مجموعة من الأشياء(الأجزاء) تربط بينها علاقة لتحقيق هدف معين.

التحليــل Analysis

يشير هذا المصطلح في مجال الحاسوب إلى عملية استقصاء وفحص النظام الحالي لفهمه وتوضيح أي غموض فيه ثم تحديد أهداف ومتطلبات تطويره أو ميكنته. ويتم ذلك عن طريق تجزئة عمليات (وظائف) النظام إلى عمليات فرعية صغيرة يسهل التعامل معها.

كمثال على ذلك نجد أن نظام المبيعات في شركة ما يمكن تحليله بأن يتجزا إلى العمليات التالية:

1. استلام الطلبية
2. تنفيذ الطلبية
3. تسليم الطلبية

وكل عملية من هذه العمليات يمكن أيضا تجزئتها إلى عمليات فرعية. على سبيل المثال يمكن تجزئة العملية الثانية (تنفيذ الطلبية) إلى العمليات الفرعية التالية:

1. تأكد من وجود الأصناف المطلوبة
2. تأكد من وضع الزبون
3. جهز قسيمة الدفع.
4. اطلب القيمة المستحقة من الزبون

التصميم Design

التصميم هو عملية تمثيل الحل باستخدام المخططات والجداول والخوارزميات ، ويكون ذلك قبل عملية التنفيذ (أي كتابة شفرة البرنامج).

ويمكن تشبيه التصميم بعملية رسم خريطة لمبنى بواسطة المهندس المعماري قبل عملية البناء.

وبعبارة أخرى، فإن التصميم هو الجسر الواصل بين تحليل النظام والتنفيذ.

التصميم الهيكلي Structured design

يعتبر التصميم هيكليا إذا استخدم أسلوب (من فوق إلى تحت top-down) أي وضعه في شكل هرمي او شجري حيث يتم تصميم النظام برمته بطريقة عامة تحتوي على جميع النظم الفرعية ، ثم يتم وضع

التفاصيل بالاستمرار في تجزئته إلى أن نصل إلى أصغر ما يمكن من مكونات (أجزاء برمجية) وهذه الهيكلية تشبه المخطط التنظيمي للشركات والمؤسسات.

1.2 مراحل إعداد منظومة

تمر عملية إعداد منظومة عادة بستة مراحل هي:

1. **مرحلة التخطيط : Planning**

خلال هذه المرحلة نحدد:

- ❖ الأهداف
- ❖ النطاق /الحدود
- ❖ المستلزمات
- ❖ الجدول الزمني
- ❖ القيودات
- ❖ .الميزانية
- ❖ الوظائف المراد أداؤها
- ❖ الإخطار المتوقعة

2. مرحلة التحليل : Analysis

في هذه المرحلة يجب تحديد ماذا ستعمل المنظومة وهذا يعني الوظائف (المتطلبات الوظيفية) التي ستؤديها المنظومة المقترحة ، وأيضا المتطلبات الغير وظيفية مثل سهولة الاستخدام والصيانة والخلو من الأخطاء .

3. مرحلة التصميم : Design

في هذه المرحلة عادة ما نبين كيف يتم تحقيق متطلبات النظام أو تمثيل الحل المقترح. ويمكن تصميم أو تمثيل الحل باستخدام واجهة المستخدم وهياكل البيانات والخوارزميات.

ويجب أن يكون التصميم كاملا بما يكفي لتنفيذه.

ملاحظة :

ترجمة المصطلح system تجدها تارة بمعنى نظام وتارة منظومة.

و الفرق بين الاثنين (في هذا الكتاب) هو أن المنظومة تشير إلى نظام تمت ميكنته، أي برمجته باستخدام الحاسوب، فمثلا نعني بمنظومة المرتبات البرنامج الذي يقوم بتجهيز المرتبات.

4. مرحلة التنفيذ Implementation

يتم في هذه المرحلة ترجمة الخوارزمية وواجهة المستخدم وقاعدة البيانات التي أعدت في مرحلة التصميم إلى شفرة برنامج بإحدى لغات البرمجة أو نظام إدارة قاعدة البيانات وإدخالها للحاسوب بغية الحصول على منظومة جاهزة للشغل.

5. مرحلة الاختبار Testing

إن الغاية من هذه المرحلة هي اكتشاف أي خطأ في المنظومة المعدة وكشف أي وظائف لم يتم إعدادها والتأكد من أن المتطلبات قد تحققت بالكامل.

وتشتمل هذه المرحلة أيضا على اختبار يجريه المستخدم قبل استلامه المنظومة والبدء في تشغيلها.

6. مرحلة الصيانة Maintenance

بعد تسليم المنظومة الى المستخدم قد تظهر بعض المشاكل مثل:

- استحداث قوانين جديدة
- تغييرات في التقنية
- أخطاء غير متوقعة لم تكتشف في مرحلة الاختبار
- وظائف جديدة يمكن إضافتها

وتسمى عملية معالجة هذا النوع من المشاكل بالصيانة أو التعمير (الدعم support).

1.3 نشاطات التحليل

فيما يلي سرد لأهم النشاطات التي نقوم بها في مرحلة التحليل :

- ❖ تجميع الحقائق أو المتطلبات
- ❖ فهم النظام الحالي
- ❖ تحديد المتطلبات للمنظومة الجديدة
- ❖ تحليل المتطلبات باستخدام أدوات التحليل
- ❖ كتابة وثيقة المتطلبات

1.4 نشاطات التصميم

النشاطات التالية تتم في مرحلة التصميم:

- ❖ تصميم معمارية النظام (تجزئة النظام)
- ❖ تصميم واجهة المستخدم
- ❖ تصميم الخوارزميات
- ❖ تصميم الملفات وقواعد البيانات (هياكل البيانات) .

1.5 أهداف التحليل والتصميم :-

- ❖ فهم النظام الحالي
- ❖ تحديد متطلبات المستخدم
- ❖ تصميم الإجراءات وقاعدة البيانات وواجهة المستخدم
- ❖ اختيار تقنية مناسبة للنظام المقترح من برمجيات وعتاد

1.6 محلل النظم System Analyst

محلل النظم هو الشخص الذي يتفاعل مع المستخدمين والمدراء لكي يفهم مشاكل النظام الحالي ويحدد ويحلل متطلبات النظام الجديد (أو المنظومة). وهو مسؤول أيضا عن تصميم النظام الجديد الذي يلبي المتطلبات وبهذا يتفاعل مع المبرمجين.

فهو عليه أن يقوم بعدد من الوظائف (المهام) مثل :

1. جمع وتحليل بيانات النظام الحالي.
2. جمع وتحليل متطلبات المنظومة الجديدة.
3. التوصيات بخصوص حلول مشاكل النظام الحالي وأي تغييرات مقترحة.

4. استعمال طرق جمع البيانات (إيجاد الحقائق) لغرض دراسة النظام الحالي وتحديد المتطلبات الجديدة.
5. عمل دراسة جدوى
6. كتابة خطة المشروع (الجدول ، تقدير التكاليف، والمستلزمات اللازمة للمشروع ، ...).
7. نمذجة النظام الحالي والجديد باستعمال أدوات وتقنيات التحليل والتصميم.
8. عمل محاضرات واجتماعات ومقابلات مع الزبون وأعضاء فريق العمل
9. كتابة التقارير والوثائق.
10. تصميم الشاشات والتقارير للمنظومة الجديدة .
11. كتابة الخوارزميات.

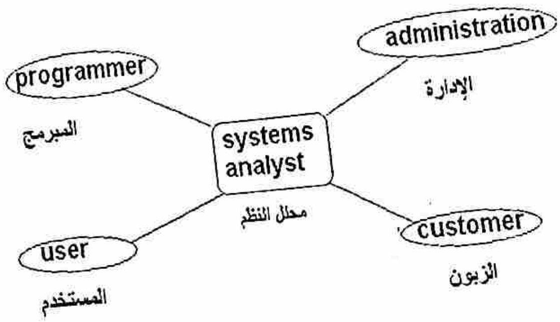
1.6.1 مـــواصفات محلل النظم

لكي يقوم محلل النظم بهذه المهام والوظائف، عليه أن يتصف بما يلي:-

1. يجب أن يكون ملما باساسيات الاعمال أي معرفة عامة بالاعمال والمهن مثل المحاسبة والتسويق والاقتصاد والإدارة.

2. استعمال مهارات الاتصال (شفوي وكتابي) للتفاعل مع المستخدم والزبون والمبرمج.
3. مستوى راقي من الابتكار والابداع في بناء منظومة متطورة.
4. قدرات تحليلية ومنطقية.
5. مهارات إدارية : مثل التخطيط والمتابعة والتوجيه.
6. مهارات تقنية : محلل النظم يجب أن يمتلك معرفة فنية جيدة ومهارات في استعمال أدوات التحليل والتصميم والبرمجة ، بالإضافة إلى المواضيع ذات العلاقة مثل : أنظمة التشغيل ولغات البرمجة والتركيب المادي للحاسوب والشبكات ...

يبين الشكل 1.6.1 موقع محلل النظم فهو الحلقة المركزية بين مستخدم النظام والزبون والمبرمج والمدير..الخ.



الشكل(**1.6.1**) محلل النظم يتعامل مع مختلف الأشخاص

1.6.3 المشاكل التي قد تواجه محللي النظم:

هناك بعض المشاكل التي قد تواجه محلل النظم أثناء تأديته لعمله مثل:

1. وجود العديد من الأدوات في السوق ، مما يسبب في عدم وجود أدوات قياسية متعارف عليها وتنوع الأدوات عن بعضها بحيث يصعب التوصية بأداة معينة.

2. ظهور تقنيات البرمجيات والعتاد تتغير بمعدل سريع ، مما يضطر محلل النظم إلى متابعة أي تطور جديد في هذا الشأن.

3. ظهور خدمات ومنتجات وقواعد ولوائح جديدة مما يلزم محلل النظم أن يكون مدركا لها.

4. العديد من المستخدمين ليس لهم دراية كافية بتقنية الحاسوب ومصطلحاته، مما يجعل عمل محلل النظم صعبا، والتخاطب مع المستخدين ليس من السهولة كما في التعامل مع التقنية.

لاحظ الفرق بين الزبون والمستخدم. فالزبون هو الذي يدفع فاتورة الحساب أما المستخدم فهو الذي يستعمل المنظومة بعد تركيبها. لاحظ أيضا أن محلل النظم يجب أن يقوم بتعريف المشكلة بوضوح وبلغة دقيقة (مستخدما النص والرسومات) وبطريقة مقبولة من كل من الزبون والمبرمج.

1.7 أدوات تحليل وتصميم النظم

مرّ فن تحليل وتصميم النظم بعدة مراحل خلال الخمسين سنة الأخيرة.

1. في البداية كان المخطط الانسيابي Flowchart هو الوسيلة الرئيسية في التحليل والتصميم.

2. بعد ذلك ومع ازدياد تعقد التطبيقات تم إنتاج أساليب جديدة وتم اتباع الأسلوب الهيكلي , والذي يستعمل أدوات هيكلية مثل

مخطط انسياب البيانات والإنجليزية المركبة، والمخطط الهيكلي....الخ.

3. ومع بداية التسعينيات ظهرت أدوات تسمى UML في التحليل والتصميم لغرض تقليص زمن إعداد المنظومات مستفيدة من أسلوب البرمجة الشيئية.

1.8 إرشادات في التحليل والتصميم

التوجيهات التالية يجب أن تؤخذ في الاعتبار عند تحليل وتصميم النظام:

تحليل وتصميم النظم عملية تكرارية (أي يتم تحسينها بعد عدة محاولات).

1. المنظومة المطلوب إعدادها هي في النهاية للمستخدم، لذلك يجب عدم تجاهله، بل يجب أن يشارك في مراحل إعدادها .

2. التوثيق مهم لجميع الناس المشاركين في إعداد المنظومة واستخدامها. لذلك يجب على محلل النظم أن يكتب بوضوح وبدقة توثيق المنظومة.

وكما يقال (الصورة تعادل الف كلمة) ، لذلك يجب على محلل النظم أن يستخدم الصورة بدلا من النص كلما أمكن ذلك للتواصل مع المستخدم. وهذه الصورة يمكن أن تتمثل في جداول أو مخططات أو شاشات.

1.9 إدارة المشروع Project Management

المشروع هو مجموعة من النشاطات حسب خطة معينة التي يجب تنفيذها للوصول إلى هدف معين.

ويجب أن يشمل المشروع :

- ❖ مدير للمشروع
- ❖ تاريخ البداية
- ❖ تاريخ النهاية
- ❖ المتطلبات
- ❖ الأهداف
- ❖ المستلزمات
- ❖ الميزانية
- ❖ حدود

ويجب أن يكون للمشروع جدول زمني ومستلزمات منذ بدايته. وبعدها نحتاج إلى مراقبة ومتابعة في تقدمه.

أما مدير المشروع فهو الشخص المسئول عن تخطيط ومتابعة وتنظيم وتوجيه وتعيين موظفين للمشروع.

ولكي ينجز هذه المهام يجب عليه استخدام الأدوات والطرق والمعايير والتقنيات الملائمة.

و يتم إعداد المنظومات عادة كمشاريع. ويتم تدشين المشروع البرمجي بواسطة:

المستخدمين

1. الزبائن

2. الإدارة

ومدير المشروع البرمجي يجب أن يكون لديه:

- ❖ خبرة محلل النظم
- ❖ مهارات تقنية وإدارية وتواصل مع الناس.
- ❖ قدرات قيادية

إدارة المشروع Project Management هي دراسة واستخدام الطرق والأدوات والمفاهيم في تخطيط ومتابعة وتنظيم وتوجيه وتوفير كوادر بشرية للمشروع البرمجي بما يحقق نجاحه.

تعتبر إدارة المشروع البرمجي مهمة للأسباب التالية:

- ❖ المشاريع عادة ما يتم أداؤها بواسطة فريق ما، وهذ الفريق يحتاج إلى أن يشرف عليه مدير

- ❖ بدون إدارة تفشل المشاريع بسبب التخطيط السيئ.والتخطيط هو احد المكونات المهمة لإدارة المشاريع.

تتكون دورة إدارة المشروع من أربعة مراحل. وكل مرحلة تتكون من نشاطات أو مهام.

المراحل الأربعة هي:

- ❖ مرحلة بدء المشروع
- ❖ مرحلة تخطيط المشروع
- ❖ مرحلة تنفيذ المشروع
- ❖ مرحلة انهاء المشروع

مثلا في مرحلة التخطيط ، إليك بعض النشاطات التي يتم إنجازها:

- ❖ اجراء دراسة الجدوى
- ❖ إعداد وثيقة خطة المشروع
- ❖ إعداد وثيقة مقترح المشروع

1.10 أسئلة للمراجعة

1. عرف :

- ▪ التحليل

- التصميم
- المشروع
- الزبون
- المستخدم

2. ما أهداف التحليل والتصميم؟
3. ما دور محلل النظم؟
4. ما دور مدير المشروع؟
5. اذكر ثلاثة ارشادات لمحلل النظم عند اجراء التحليل والتصميم.
6. ما مراحل دورة حياة إعداد المنظومة؟
7. ما الفرق بين : التحليل والتصميم ؟
8. ما هي إدارة المشاريع؟ وما هي مراحلها؟
9. لماذا نحتاج إلى إدارة المشاريع في إعداد المشاريع؟

ما هي مؤهلات مدير المشروع البرمجي ؟

2

Information Systems

نظم المعلومات

1. مقدمة
2. ماهو النظام
3. النظام الفوقي والنظام الفرعي
4. نظم المعلومات
5. المعلومات الحاسوبية
6. نظم معلومات الأعمال
7. تقنية المعلومات
8. أسئلة مراجعة

2.1 مقدمـــة

نحـن نعـيش في عـالم مـن الـنظم الـتي تلعـب دورا مهمـا في حياتنـا اليومية، ونلمسها تقريبا في كل مكان. كأمثلة على هذه الأنظمة : النظام السياسي، النظام الاقتصادي، النظام الاجتماعي....الخ.

وكـثيرا مـا نـستخدم كلمـة النظـام (مـثلا النظـام الكهربـائي، نظـام الحاسوب، ..الخ) ولكن قليل منـا مـن يستطيع أن يعطـي تعريفـا واضحا لما هو النظام؟

2.2 مـــا هو النظـــام؟

النظام system هو "مجموعة من الكينونات entities أو الأشياء ذات العلاقة التي تربط بينها بحيث تكوّن وحدة واحدة لغرض تحقيق هدف معين".

وبعبارة أخرى يمكن تعريف النظام بأنه:

*"مجموعة من الأجزاء **parts** ذات علاقة ما ، تعمل معا لتحقيق غرض أو هدف معين. هذا الهدف أو هذه الأهداف يمكن تحقيقها بالتفاعل مع البيئة وذلك بأداء الوظائف المطلوبة للنظام".*

خصائص النظـــام System Characteristics

لاحظ أن أي نظام يمتلك الآتي :

- **هـــدف أو غاية Gool** :- الغاية من اعداد النظام

❖ **المدخلات** Input : الأشياء التي تدخل النظام مثل البيانات.

❖ **المخرجات** Output : الأشياء التي تخرج من النظام مثل المعلومات.

❖ **الحدود** Boundaries : وهي تحيط بالنظام وتفصله عن البيئة المحيطة به، وتحدد ما يشمله النظام وما هو خارج عنه.

❖ **البيئة** Environment : كينونات خارجية تتفاعل مع النظام مثل : اشخاص ، برامج ، نظم، ... الخ.

❖ **المكونات** Components : يمكن أن تكون بيانات، أشخاص ، عتاد ، برمجيات، إجراءات، ... الخ.

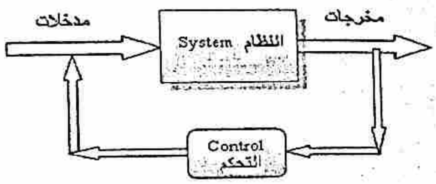
❖ **الوظائف** Functions : العمليات التي يقوم بها النظام.

❖ **القيود** Constraints: الضوابط أو الحصر (مثل الزمن والتكلفة).

❖ **الاسترجاع** Feedback : بعض الأنظمة (كما في الشكل2.2.1) تتمتع بعامل التحكم، وذلك عن طريق التغذية الاسترجاعية، التي

تقوم بقياس المخرجات باستخدام معايير معينة، وبناء على ذلك يتم تعديل المدخلات أو المعالجة ، لغرض تحسين المخرجات.

في تحليل النظم ، يتم الاستفادة من التغذية الاسترجاعية لتوضيح متطلبات المستخدم user requirements.



الشكل(2.2.1) نظام استرجاعي

النظم المغلقة والمفتوحة: Open \ Close Systems

النظام إما أن يكون مغلقا أو مفتوحا.

النظام المغلق closed system هو النظام المعزول عن البيئة المحيطة به.

على سبيل المثال، فإن التفاعلات الكيميائية التي تحدث داخل أنبوبة محكمة الإغلاق دون أي تسرب تعتبر نظاما مغلقا.

النظام المفتوح open system هو الذي يتفاعل مع البيئة المحيطة به. بعبارة أخرى، النظام المفتوح يجب أن يكون له بيئة ، ويتأثر ببعض العوامل الخارجية.

يعتبر النظام المصرفي نظاما مفتوحا لأن المكونات الخارجية مثل الزبون والموظف والصراف هم جزء من البيئة يتفاعلون مع النظام.

ملاحظة :
معظم المنظومات البرمجية تعتبر من النوع المفتوح لأنها تتفاعل مع البيئة.

2.3 أمثلة على النظم

1- نظام حاسوبي Computer System

مجموعة من الأجهزة Hardware (العتاد) والبرمجيات Software تحت نوع معين من التحكم لمعالجة بيانات وإنتاج مخرجات.

نظام اتصالات Communication System
مجموعة من الوحدات التي تمثل المعلومات وتبثها من نقطة إلى أخرى.

3- نظام تسويق Marketing System

مجموعة من الناس والبضائع والمعدات والإجراءات تعد أو تنتج وتوزع بضاعة أو خدمات للمستخدم.

2- نظام الحجز في الخطوط الجوية

نظام يتكون من:

- ❖ مسافرين
- ❖ رحلات
- ❖ طائرات
- ❖ شحن
- ❖ موظفين
- ❖ اجراءات

لغرض نقل الركاب والبضائع.

2- النظام المصرفي Banking System

مجموعة متكونة من :

- ❖ إجراءات
- ❖ يد عاملة
- ❖ زبائن
- ❖ عملة

لغرض تقديم خدمات مصرفية مثل الحسابات الجارية وحسابات التوفير والقروض.

3- نظام مستشفى Hospital System

هو مجموعة تتكون من :

- ❖ مستلزمات
- ❖ نزلاء
- ❖ غرف
- ❖ معدات
- ❖ طاقم طبي
- ❖ إجراءات

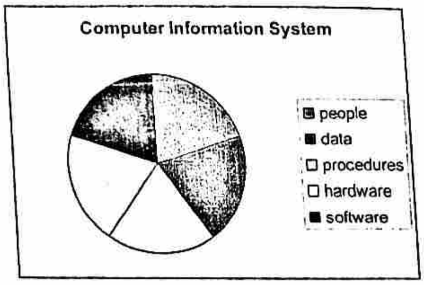
لغرض تقديم خدمات صحية.

4- نظـــام المعلومـــات الحاسوبي computer information system

يتكون نظام المعلومات الحاسوبي من

- الأشخاص People
- البيانات data
- الاجراءات procedures
- العتاد hardware
- البرمجيات software

كما في الشكل التالي:



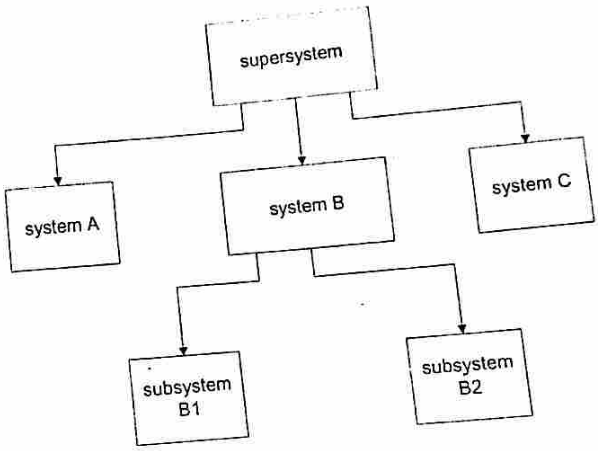
الشكل(2.3.1) عناصر نظام معلومات حاسوبي

2.4 النظـــم الفوقيـــة supersystems والنظـــم الفرعيـــة subsystems

كل نظام هو نظام فرعي (أي جزء) من نظام أكبر.

يسمى النظام الكبير الذي يحتوي على الأنظمة الفرعية بالنظام الفوقي.

في الشكل 2.4.1 نجد ثلاثة أنظمة هي A و B و C تنتمي إلى نظام فوقي واحد. لاحظ في هذا الشكل أن النظام B له نظامان فرعيان هما B1 و B2.



الشكل (2.4.1) النظام الفوقي والفرعي

مثــــال:

قسم المبيعات في المؤسسة هو نظام فرعي من تلك المؤسسة.
ويمكن أن تحتوي المؤسسة على نظم فرعية أخرى مثل:

- ❖ قسم الإنتاج
- ❖ قسم الهندسة

- ❖ قسم الحسابات

لاحظ أن المؤسسة هي جزء من صناعة ما. لذلك فهي تعتبر أيضا نظام فرعي من نظام أكبر (الصناعة).

من الأسهل دائما والأيسر أن نبسط الدراسة بتقسيم النظام إلى وحدات.

الوحدة هي في الحقيقة أصغر نظام فرعي (أي لا يمكن تقسيمه إلى وحدات أصغرمنها).

التفاعل بين الأنظمة الفرعية يتكون من تبادل بين المدخلات والمخرجات (أو المواد).

من معرفة نتائج الإدخال والإخراج لعمليات نظام فرعي معين، يمكننا وصل نظام فرعي بآخر.

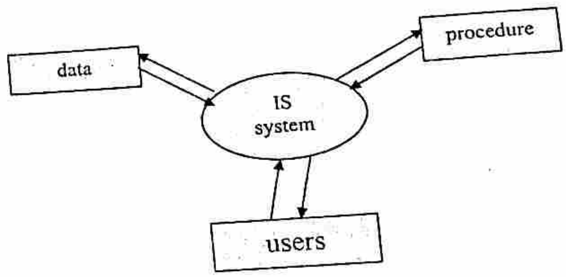
2.5 نظم المعلومات

Information System (IS)

نظام المعلومات هو نظام يتكون من :

- ❖ المستخدمين users
- ❖ الإجراءات procedures
- ❖ البيانات data

كما مبين بالشكل (2.5.1) .



الشكل2.5.1 مكونات نظام المعلومات

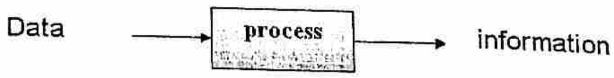
والإجراء هو وصف كتابي (أو مخطط) للأعمال التي يجب إنجازها لأداء مهمة معينة.

المعلومات information هي المخرجات output التي نتحصل عليها من معالجة البيانات data كما مبين بالشكل (2.5.1).

ويجب أن تكون المعلومات :

- ❖ دقيقة
- ❖ كاملة
- ❖ متوفرة عند الحاجة

- ❖ ذات علاقة بالمسألة الحالية



الشكل(2.5.2) المعلومات و البيانات

لاحظ أن المصطلح (معالجة process) يعني تحويل البيانات إلى معلومات.

وظائف نظـــام المعلومـــات

يجب على نظام المعلومات أن يؤدي الوظائف التالية لإنتاج المعلومات:

- ❖ تجميع البيانات وتحليلها
- ❖ معالجة البيانات
- ❖ إدارة البيانات
- ❖ أمن البيانات والتحكم بها
- ❖ إنتاج المعلومات

أهـــداف نظام المعلومـــات

يسعى نظام المعلومات الى تحقيق الأهداف التالية:

- الحصول على المعلومات في الوقت المناسب عند اللزوم
- تحسين عملية اتخاذ القرار للإدارة
- الاستخدام الأفضل للقوى العاملة
- التقليل من عملية تكرار البيانات

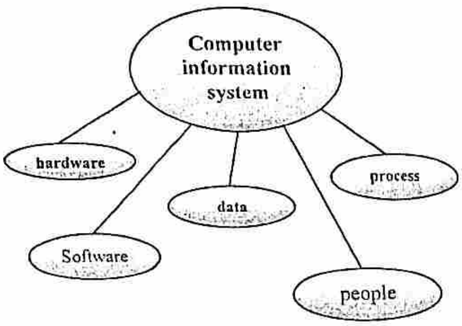
2.6 نظـــام المعلومـــات الحاسوبي

CIS = Computer Information System

يعرّف نظام المعلومات الحاسوبي بأنه نظام يتكون من:

- بيانات data
- برمجيات software
- عتاد hardware
- العنصر البشري people
- معالجة processing

كما مبين في الشكل 2.6.1 :



الشكل(2.6.1) مكونات نظام معلومات حاسوبي

ويمكن توضيح كل عنصر من هذا النظام حسب الآتي:

1- البرمجيـــات Software

البرمجيات هي الكيان المعنوي لنظام الحاسوب ، وتنقسم الى:

❖ ***برمجيات تطبيقية*** Application Software مثل برمجيات الرواتب والتحكم في المخزون

❖ ***برمجيات النظم System Software*** مثل :-

- نظام التشغيل operating system
- المترجمات compilers

- النسخ الاحتياطية back-up
- الحماية من الفيروسات anti-virus

2-العتـــاد Hardware

العتـاد : يـشمل الحواسـيب ومعـدات الاتـصال والـشبكات والطابعات والماسحات ...الخ.

3-العنـصر البـشري People : وهـم الزبـائن والموظفـون والمـوردون ومستخدمو النظام.....الخ.

4-البيـانـــات Data

البيانـات : هي سيل من الحقائق الخام مثل الأسماء والعناوين وهي تمثل الأحداث قبل معالجتها إلى معلومات مفيدة.

5-المعالجـــة Processing

المعالجة : هي نهج في طريقة إنجاز المهام. وتسمى أيضا الاجراءات procedures

2.7 نظم معلومــات الأعمال

Business Information System (BIS)

هـو نظـام معلومـات لمؤسـسة تجاريـة تقـوم بإنتـاج منتجـات أو تقـديم خدمات أو كلاهما وتستخدم الحاسوب في أداء أعمالها.

وتـشمل الخـدمات في هـذه الأيـام التجـارة الإلكترونيـة والمـصارف الإلكترونية عن طريق الإنترنت.

ويمكن تقسيم نظم معلومات الأعمال إلى:

(1) منظومـــات معالجة المعاملات

Transaction processing systems (TPS)

وهي المنظومات السائدة والاكثر استخداما والتي نجدها في المصارف وشركات الطيران والمستشفيات والمحلات التجارية وغيرها. وتسمى أيضا منظومات العمليات.

(2) منظومــة إدارة المعلومات

Management Information System (MIS)

هذا النوع من المنظومات يساعد المد راء في اتخاذ قرارت أفضل. وتستخدم البيانات المتوفرة من خلال منظومة العمليات (TPS) وتحول هده البيانات إلى معلومات يحتاجها المدير في صناعة القرار وتكون في شكل تقارير تطبع على الطابعة ليستفيد بها المدير في اتخاذ القرارات.

مثـال : باسـتخدام بيانـات منظومـة المبيعـات يمكـن لمنظومـة إدارة المعلومات أن تطبع تقرير يشير إلي الأصناف الأكثر رواجا(نقطة إعادة الطلب) والأصناف البطيئة الحركة . وبهده الطريقة يستطيع المدير من

خلال التقارير أن يتصل بقسم المشتريات لجلب أصناف جديدة إلى المخزن عند قرب نفاد الأصناف المشار إليها.

(3) المنظومـــات الخبيرة Expert Systems

تـــسمى هـــذه المنظومـــات أيـــضا بمنظومـــات إدارة المعرفـــة Knowledge Management System (KMS)

يوصف النظام بأنه نظام خبير إذا كان يستعمل الذكاء الاصطناعي لتوفير معلومات مفيدة .

وبعد انتشار استخدام الإنترنت أصبحت المنظومات الخبيرة يشار إليها باسم منظومات إدارة المعرفة لأنها تستخدم الإنترنت للبحث عن المعرفة.

فمنظومة إدارة المعرفة تملك قاعدة بيانات كبيرة تمكّن المستخدمين من الوصول الى المعلومة طبقا لمعيار معين.

وكمثال على منظومة المعرفة استعمال محرك البحث في الإنترنت للبحث عن أماكن شاغرة في الفنادق أو الطيران.

ومثال آخر على المنظومات الخبيرة نجده في منظومة التشخيص الطبي التي تدعم الطاقم الطبي للحصول على تشخيص للمرضى.

(4) منظومات تحسين إنتاجية المستخدم (الموظف)

Employee productivity enhancement systems وتسمى ايضا منظومات ميكنة المكاتب Office Automation.

و هذه المنظومات تستخدم من قبل مختلف موظفي الحاسوب لمساعدتهم في أداء أعمالهم الروتينية (مثل الطباعة) بكفاءة وفعالية.

ومن أمثلة هذه المنظومات:

- ❖ البريد الإلكتروني
- ❖ معالجة النصوص
- ❖ الجداول الإلكترونية
- ❖ قواعد البيانات
- ❖ المؤتمرات عن بعد

وأشهر منظومة في هذا المجال هي : **MS-OFFICE** من إعداد شركة مايكروسوفت Microsoft .

(5) منظومـــات دعم القرار

Decision Support System (DSS)

هذا النوع من الأنظمة يستخدم لمساعدة المدراء في اتخاذ قرارات أفضل لصالح المؤسسة.

الفرق بين MIS و DSS هو أن MIS ينتج تقارير (مقتبسة من TPS) لغرض استخدامها من قبل مدراء كأدوات مساعدة لاتخاذ القرارات، بينما DSS تستخدم لتوفير بيئة تفاعلية مع الحاسوب عن طريق الشاشة بدلا من الطابعة للتعامل مع البيانات والحصول على المعلومات الضرورية لاتخاذ القرارات.

لاحظ أن البيانات في هذه المنظومة غير حقيقية فمثلا يمكن أن يدخل المدير سعر صنف إلى المنظومة ويريد بيع 1000 صنف مثلا فتقوم المنظومة بحساب الربح ويقارن ذلك عندما يبيع10000 صنف فتحسب المنظومة الأرباح وهكذا مع باقي الأصناف.

لهذا تعتبر منظومة DSS منظومة محاكاة تعطى النتائج قبل حصولها على أرض الواقع وبهذا تساعد في اتخاذ القرار.

و مكونات DSS هي قاعدة البيانات وواجهة المستخدم وأدوات رسم .

2.8 تقنيـــة المعلومــات

Information Technology (IT)

يمكن تعريف تقنية المعلومات بأنها مزيج من البرمجيات والعتاد والاتصالات التي تدعم التطبيقات وتحسن العملية الإنتاجية لمساعدة المدراء في اتخاذ القرارات السليمة داخل المؤسسة.

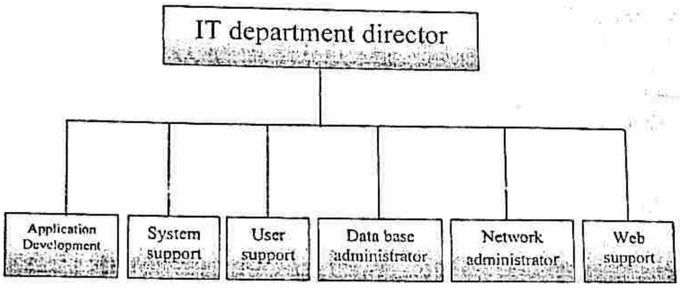
لقد بدأ قسم تقنية المعلومات في الظهور في الشركات والمؤسسات بعد انتشار استخدام الشبكات والوسائط المتعددة والإنترنت.

هذا القسم يساعد المؤسسة (مثل الشركة والمصنع والمصرف ...الخ) في تخزين البيانات ومعالجته ونقلها الكترونيا عبر الشبكات و بتعبير آخر ، يقوم بالإشراف على نظام معلومات المؤسسة.

وقد حل قسم تقنية المعلومات IT محل الأقسام القديمة ذات العلاقة بخدمات الحاسوب مثل:

1. قسم معالجة البيانات Data Processing (DP)
2. قسم الحاسوب Computer Department
3. قسم المعلومات Information Department

ويمكن تقسيم الوحدات الأساسية لقسم تقنية المعلومات كما في الشكل 2.8.1:



الشكل(2.8.1) مخطط تنظيمي لقسم تقنية المعلومات

حيث _IT Director_ تعنى مدير قسم تقنية المعلومات ، والوحدات التابعة لهذا القسم تشمل:

وحـــدات قسم تقنيـــة المعلومـــات :

1. وحـــدة إعـــداد التطبيـــقات Applications Development Unit

تتكون وحدة إعداد التطبيقات من مدير ومصممين ومحللين ومبرمجين وكتبة تقنيين ..الخ وذلك لغرض إعداد(منظومات) البرمجيات التطبيقية.

2. وحـــدة دعم النظام System Support Unit

توفر هذه الوحدة البرمجيات والعتاد اللازم لدعم المؤسسة بالكامل وهذه الوحدة تدعم مرحلة الصيانة بعد البدء في تشغيل المنظومة على أرض الواقع.

ومـن خـدمات هـذه الوحـدة تركيـب وصـيانة نظـم قواعـد البيانـات (مثـل أوراكل Oracle و إس كيو إل SQL) ونظام التشغيل (مثل ويندوز windows أو لينوكس Linux) وبرمجيات الاتصالات.

3. وحـــدة دعم المستخدم User Support Unit

وهذه الوحدة تتعامل مع المستخدم مباشرة.

والوظائف المنوطة بهذه الوحدة مايلي:-

- توفير المعلومات التقنية للمستخدمين إلى جانب التدريب . وتسمى هذه الوظيفة أيضا بمكتب المساعدة help desk . وخدمات التدريب التي تقدمها هذه الوحدة للمستخدمين والمد راء هي :
  1. تشغيل منظومة جديدة
  2. معالجة النصوص
  3. الجداول الإلكترونية

- مساعدة المستخدمين في حل أي مشكلة تواجههم عند استعمال المنظومة الجديدة .

4-وحـــدة إدارة قاعدة البيانات Data base Unit

تقوم وحدة إدارة قاعدة البيانات بالمهام الآتية :

- تصميم قاعدة البيانات
- آمن وسلامة قاعدة البيانات
- إدارة قاعدة البيانات
- صلاحيات الدخول الخاصة بالمستخدمين
- عمل النسخ الاحتياطية .

5-وحـــدة إدارة الشبكة Network unit

المهام المنوطة بهذه الوحدة هي :

- تركيب ومتابعة تطبيقات الشبكة
- تمكين الولوج إلى الشبكة لعدد من المشغلين
- صيانة البرمجيات والعتاد الخاص بالشبكة
- دعم الشبكة
- أمن وسلامة الشبكة

وحـــدة دعـــم الشبكة العنكبوتية Web Support Unit

تقوم وحدة دعم الشبكة العنكبوتية بالمهام الآتية:

- ❖ دعم الإنترنت والإنترانت
- ❖ تصميم وبناء صفحات الإنترنت
- ❖ تدريب المستخدمين على استخدام الإنترنت
- ❖ ربط تطبيقات الإنترنت بالمنظومة الحالية للمؤسسة

الفنيون في هذه الوحدة يسمون اخصائيي الشبكة العنكبوتية

مــلاحظة:

الانترانت **Intranet** هي شبكة معلومات خاصة بمؤسسة معينة تقدم تقريبا نفس خدمات الانترنت **Internet** مثل البريد الالكتروني والمؤتمرات الالكترونية.

2.9 أسئلـــة للمراجعـــة

1. ما هو النظام؟ ما هي خصائصه؟ اذكر 3 أمثلة للنظام.
2. كيف يعمل نظام التغذية الاسترجاعية؟.
3. قارن بين النظام المفتوح والنظام المغلق.
4. اذكر مثالا على نظام مع تحديد نظام فوقي ونظام فرعي له.
5. عرف نظام المعلومات واذكر مكوناته.
6. ما هي وظائف وأهداف نظام المعلومات؟
7. ما هي مكونات نظام معلومات حاسوبي؟
8. عرف ما هو نظام معلومات الأعمال واذكر 5 أنواع لهذا النظام.
9. لماذا يجب على مؤسسة ما أن يكون لديها قسم تقنية المعلومات؟
10. ماذا تعني ITD ؟ وما هي وحداتها؟
11. اذكر 4 صفات يجب في رأيك أن تتمتع بها المعلومات حتى تكون ذات فائدة.
12. أغلب البرمجيات الموجودة في السوق يمكن تصنيفها إلى صنفين. ما هما؟
13. ماذا تعني الاختصارات التالية:

❖ MIS

- ES ❖
- BIS ❖
- KMS ❖

14. اذكر مثالا على

- Expert system ❖
- User Productivity ❖
Enhancement System.

3

دورة حياة إعداد المنظومة
System Development Life Cycle (SDLC)

1. مقدمة
2. مرحلة التخطيط
3. مرحلة التحليل
4. مرحلة التصميم
5. مرحلة التنفيذ
6. مرحلة الاختبار
7. مرحلة الصيانة
8. مراجعة البرمجيات
9. التوثيق
10. أسئلة للمراجعة

3.1 مقدمـــة

من المعلوم أن منظومة المعلومات يجب أن تكون ذات اعتمادية وكفاءة عالية حتى لا تحدث مشاكل كبيرة بسبب الأخطاء المكلفة.

ومن الأمثلة المعروفة على هذا النوع من المشاكل الصعبة ما يعرف بمشكلة سنة 2000 التي نتجت من كون المنظومات المعدة في الستينيات والسبعينيات من القرن العشرين كانت تستخدم خانتين فقط للسنة بدلا من 4 خانات. أي على سبيل المثال سنة 46 تعني سنة 1946.

عندما اقتربت سنة 2000 أصبح جليا أن هذه المنظومات ستفشل لأن الخانتين سترمز للسنة بصفرين 2000. والحل الوحيد لهذه المشكلة كان إعادة كتابة البرامج مما كلف بلايين الدولارات.

ولو تم استخدام أسلوب سليم في تحليل النظم ما كانت هذه المشكلة لتحدث لأن التحليل سيظهر المدة الزمنية التي ستعيشها المنظومة.

لذلك فإن المنظومات يجب أن يتم إعدادها بطريقة منظمة باستخدام منهجية(طريقة) سليمة.

ويوجد ثلاثة طرق(منهجيات) تستخدم في تحليل وتصميم نظم المعلومات هي:

1. الطريقـة غير الهيكلية unstructured

وهي طريقة قديمة في إعداد المنظومات بدأت في الستينيات مستخدمة المخططات الانسيابية flow charts كوسيلة للتحليل والتصميم. وهذه الطريقة لم تعد تستخدم حاليا إلا في تصميم النظم البسيطة.

2. الطريقـة الهيكلية structured

بدأت هذه الطريقة في السبعينيات باستخدام أدوات وتقنيات هيكلية.

وهي تناسب إعداد المنظومات الكبيرة والمعقدة نظرا لأنها تقسم هذه المنظومات إلى أجزاء صغيرة مستقلة.

كمثال على هذه الأدوات في هذه الطريقة:

- مخطط انسياب البيانات (Data Flow Diagram DFD)
- المخطط الهيكلي Stuctured Chart

3. الطريقـة الشيئية (الكائنية) object-oriented:

وهي تستعمل مفاهيم متقدمة مثل إعادة استعمال الأجزاء البرمجية لتقليص وقت إعداد المنظومات. وبعض الأدوات في هذه الطريقة هي مخطط حالة الاستعمال Use Case Diagram ومخطط الفصيلة Class Diagram...الخ.

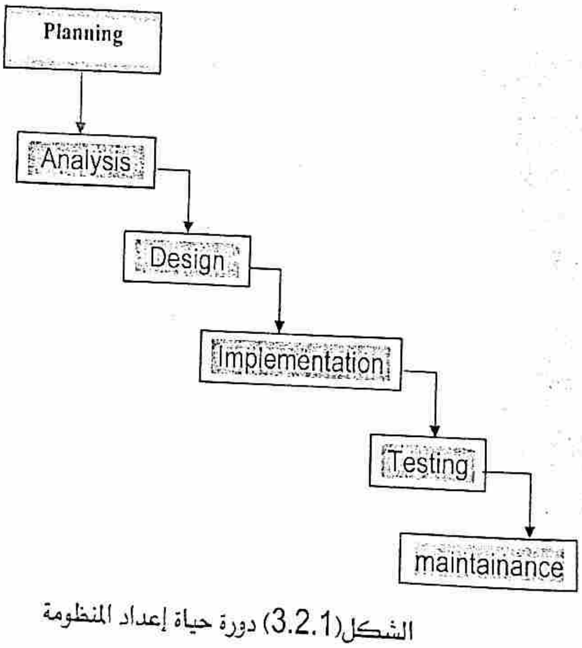
3.2 دورة حيـــاة إعداد المنظومة

System develpoment life cycle

سندرس في هذا الباب بشيء من التفصيل دورة حياة إعداد المنظومة.

إن أبسط نموذج لدورة حياة أعداد منظومة هو ما يعرف بالنموذج التتابعي أو النموذج التدفقي الذي يعني أننا نقوم بإعداد المنظومة في تسلسل مرحلة بعد مرحلة.

يتكون هذا النموذج كما مبين بالشكل 3.2.1 من عدد من المراحل بحيث تعرف كل مرحلة بأنها عبارة عن عدد من الأنشطة يجب أن تنجز في فترة زمنية .



الشكل(3.2.1) دورة حياة إعداد المنظومة

ويمكن تقسيم هذا النموذج كما هو مبين بالجدول التالي:

مراحل ونشاطات دورة حياة إعداد المنظومة

نشاطات	المرحلة
طلب المستخدم دراسة الجدوى خطة المشروع مقترح المشروع	التخطيط Planning
جمع الحقائق والمتطلبات تحليل المتطلبات	التحليل Analysis
التصميم المعماري تصميم واجهة المستخدم تصميم البيانات تصميم الاجزاء البلرمجية(الخوارزميات)	Design التصميم
التشفير اكتشاف الأخطاء اختبار الوحدة	التنفيذ Implementaion
اختبار التكامل اختبار النظام اختبار القبول	Testing الاختبار

التحسين التكيف مع التقنية تصحيح الاخطاء إعادة الهندسة	الصيانة manitainance

لاحظ في هذا النموذج أن:

1. كل مرحلة تحتاج إلى مستلزمات (برمجيات / عتاد / بشر) لإكمال كل عملياتها.
2. كل مرحلة تحتوي على مدخلات ومعالجة ومخرجات.
3. يجب اجراء مراجعة في نهاية كل مرحلة.
4. يجب إعداد وثيقة المواصفات ومراجعتها في نهاية كل مرحلة.
5. يجب التحقق من اكتمال وصحة ووضوح وثيقة المواصفات في كل مرحلة.

في هذا الباب سندرس كل مرحلة من مراحل هذا النموذج بشيء من التفصيل.

3.3 مرحلة التخطيط Planning Phase

تحتوي هذه المرحلة على أربعة أنشطة رئيسية:

- ❖ طلب المستخدم
- ❖ دراسة الجدوى

❖ خطة المشروع
❖ مقترح المشروع

طلب المستخدم User Request

تبدأ المشاريع عادة بسبب حاجة ملموسة من المستخدم، أو نتيجة مشكلة جدية تؤثر على عمليات مؤسسة ما.
وتستخدم عدة مؤسسات نموذجا خاصا لطلب المستخدم.
ويملأ المستخدم النموذج لغرض تحديد المشاكل الموجودة في النظام الحالي وتعريف الحدود والأهداف المرجوة من النظام المقترح.
في هذه النموذج، يمكن أن يطلب المستخدم واحدا من الاختيارات التالية:

❖ نظام جديد
❖ تحسين النظام الحالي
❖ تصحيح أخطاء المنظومة الحالية

نقدم هنا مثالا بسيطا لنموذج طلب المستخدم. هذا النموذج يتم تسليمه للإدارة لاتخاذ القرار بخصوص البدء بالنشاط التالي وذلك بتوظيف محلل نظم أو شركة استشارية لإجراء دراسة جدوى .
والشكل التالي يبين نموذج لطلب المستخدم مرسل إلى الإدارة العليا لغرض إعداد منظومة جديدة أو تصحيح النظام الحالي.

من : رئيس قسم المبيعات
إلى : المدير العام لشركة ABC
الموضوع : نظام المبيعات
المطلوب☹ أختر واحد من الأتي:
❖منظومة جديدة ❖تصحيح المنظومة ❖تحسين المنظومة

وصف النظام الحالي ومشاكله:
يقوم قسم المبيعات ببيع الأثاث.
حيث يدفع الزبون الفاتورة إما نقدا أو بالدين.
ويستقبل القسم أكثر من 50 زبون في اليوم الواحد منهم 50% تقريبا الدفع نقدا.
ويتوقع القسم زيادة في المبيعات نظرا لجودة المنتجات التي تقدم للزبائن.
ومع ذلك أود أن أفيدكم بأننا في حاجة لتحسين أداء قسم المبيعات.
فالعمليات في هذا القسم تجري يدويا.
والنظام الحالي يعاني بعض المشاكل في الحصول على المعلومة عند الحاجة إليها.
ونحن نود أن نطور خدماتنا والتفوق على المنافسة في السوق.

ويمكن أداء ذلك بربط جميع فروع المبيعات بمنظومة موحدة باستخدام الحاسوب والشبكات والانترنت.وبهذه الطريقة نحصل على رضا الزبائن الحاليين ونجلب زبائن جدد.

وفي حالة قبول هذا الطلب من الإدارة العليا يتم تكليف محلل نظم لإعداد دراسة جدوى بهذا المشروع الجديد.

دراسة الجدوى Feasibility Study

تتم هذه الدراسة لأغلب المشاريع المتوسطة والكبيرة الحجم. وهي تتكون من :

- ❖ الجدوى الاقتصادية Economic feasibility
- ❖ الجدوى الفنية Technical feasibility
- ❖ الجدوى التشغيلية Operational feasibility

1- الجدوى الاقتصادية Economic Feasibilty

هي تحليل التكاليف والمزايا لمعرفة ما إذا كانت المزايا تفوق التكاليف المتوقعة.

ولتحقيق هذا الهدف يقوم محلل النظم بإعداد الجدول الذى يقارن التكايف التقديرية مع العائد أو الفوائد المتوقعة حسب التالي:

تحليل التكاليف/الفوائد

1. الفوائد المتوقعة :

1. تحسين خدمات ( )
2. تقليص التأخير ( )
3. تحسين الرضا عن العمل ( )
4. اتخاذ قرارات أفضل ( )
5. تحسين صورة الشركة ( )
6. إعداد موقع وب ( )

2. التكاليف التقديرية

1. تكاليف العتاد والمعدات ----------
2. تكاليف البرمجيات ----------
3. تكاليف العنصر البشرى ----------
4. تكاليف التدريب ----------
5. تكاليف التجهيزات ----------
6. تكاليف الاستشارات ----------
7. مصاريف أخرى ----------

المجموع ----------------------

وتعتمد الجدوى الاقتصادية بشكل كبير على عامل الزمن.

وهذا يعني أن مدير المشروع يجب عليه تقدير الوقت اللازم لكي نبدأ في الحصول على عائد يفوق التكاليف.

والجدول التالي مثال لمشروع يكلف 3000 دينار ويمكن استخدامه لغرض تقدير:

- ❖ الزمن المطلوب.
- ❖ التكاليف المتوقعة
- ❖ العائد المتوقع
- ❖ انعائد التراكمي

Year السنة	Expected Cost التكلفة المتوقعة	Expected reward العائد المتوقع	Accumulative reward العائد التراكمي
1	3000	0	0
2	0	900	900
3	0	1700	2600
4	0	1000	3600
5	0	2000	5600

لاحظ في هذا الجدول أن العائد التراكمي في السنة الرابعة يزيد عن التكلفة مما يدل على الجدوى الاقتصادية. ولكن هذا العائد قد يكون مرضيا لبعض المشاريع وليس البعض الآخر. القرار النهائي يعتمد على عوامل متعددة ولاحظ أنه في السنة الرابعة يكون العائد أكبر من التكاليف أي أن المشروع سيجني أرباح بعد أربع سنوات من التشغيل.

2. الجدوى الفنية (التقنية) Technincal feasibility

دراسة الجدوى الفنية تجيب عن الأسئلة التالية:

- ❖ هل للشركة المعدات والبرمجيات والشبكة اللازمة للمشروع ؟
- ❖ إذا كان ذلك غير متوفر ، هل يمكن الحصول على هذه المستلزمات بسهولة؟
- ❖ هل للشركة الخبرات الفنية اللازمة؟
- ❖ إن كان غير ذلك فهل يمكن توفيرها
- ❖ هل سيتمكن النظام من معالجة حجم المعاملات المتزايد في المستقبل؟

إذا كانت الإجابة لهذه الأسئلة بنعم ، فإن النظام المقترح يعتبر ذا جدوى فنية.

3. الجدوى التشغيلية Operational Feasibility

هي دراسة تجيب عن الأسئلة التالية:

- ❖ هل ستكون المنظومة بعد إعدادها سهلة الاستخدام؟
- ❖ هل ستعمل المنظومة بكفاءة؟
- ❖ هل سيدعم المستخدمون والمدراء المشروع الجديد ، وهل سيكون لديهم الشعور بأن المنظومة الجديدة ستحقق احتياجاتهم؟
- ❖ هل سيتم قبول المنظومة بعد استلامها من قبل المستخدم؟
- ❖ هل ستتطلب المنظومة الجديدة تدريبا؟ وهل سيتم التدريب قبل عملية التسليم؟

إذا كانت الاجابة بنعم لهذه الاسئلة ، فإن النظام ذو جدوى تشغيلية .

يمكن أن نستخلص أن دراسة الجدوى عبارة عن تحليل مبدئي يتم قبل بداية المشروع وينتج عن هذه الدراسة تقرير (يحتوي على مجموعة بنود) يحدد ما إذا كانت المنظومة المقترحة مجدي أعدادها من النواحي الاقتصادية والتشغيلية والفنية.

والجدول التالي يبين محتويات هذا التقرير:

1- اسم المشروع
2- وصف النظام الحالي
3- الاهداف والفوائد المتوقعة
4- الحدود
5- الافتراضات / الشروط
6- حلول بديلة (لكل حل) :
   - الخطوط العريضة للنظام
   - الجدول الزمني
   - القيود
   - المستلزمات
   - المزايا والعيوب
   - الفوائد / التكاليف
7- التوصيات والخلاصة
8- الملاحق

وثيقة خطة المشروع Project plan document

يجب أن تعد هذه الوثيقة بعد الموافقة على وثيقة دراسة الجدوى واختيار الحل المرغوب.

وهي قد تشمل الآتي:

- ملخص المشروع
- جدول المشروع

- قيود المشروع
- الأهداف المرجوة
- الوظائف المطلوبة
- نطاق المشروع
- التكلفة التقديرية
- المستلزمات المطلوبة
- الأدوات والفنيات التي ستستخدم في التحليل والتصميم.
- خصائص الجودة المطلوبة
- تحليل الاخطار

ملاحظات:

❖ في الباب التالي سندرس أدوات التخطيط مثل مخطط غانت Gantt chart وجدول الخط الزمني...الخ.

❖ وسيتم التعرض إلي تحليل المخاطر

وثيقة مقترح المشروع Project Proposal

يجب أن تعرض هذه الوثيقة على الإدارة العليا للاعتمادوالموافقة .
ويجب أن تشمل هذه الوثيقة البنود التالية:

❖ وصف المشكلة

❖ مشاكل النظام الحالي

❖ شرح للنظام المقترح وتبرير اختيار النظام مقارنة بين الحلول البديلة.

❖ الجدوى الاقتصادية والفنية والتشغيلية للنظام المقترح.

❖ الأدوات التي ستستخدم في المشروع.

❖ الأفراد ودورهم في المشروع.

❖ تأثير المشروع المقترح على عمليات النظام الحالي.

❖ نسخة من خطة المشروع التي تم عرضها في البند السابق.

وبعد مرحلة التخطيط سنشرع في المرحلة اللاحقة للتخطيط الا وهي مرحلة التحليل.

3.4 مرحلة التحليل Analysis Phase

ان الهدف الرئيسي من التحليل هو تعريف متطلبات المستخدم وبتعبير آخر يجب أن نحدد في هذه المرحلة ما هو متوقع من النظام المقترح حتى نحقق متطلبات المستخدم.

ويمكن أن نقسم متطلبات المستخدم إلى ما يلي:

❖ متطلبات وظيفية : الوظائف التي ستؤديها المنظومة.

❖ متطلبات الأداء : ما هي السرعة المطلوبة من المنظومة لإنجاز الوظائف؟ ويقاس هذا الأداء بوقت الاستجابة.

❖ متطلبات الإدخال : وهي البيانات التي تعطى للمنظومة لغرض المعالجة.

- ❖ متطلبات الإخراج : هي المعلومات المتحصل عليها من المنظومة بعد معالجة المدخلات. وقد تأخذ هذه المخرجات شكل التقارير أو الاستعلامات.
- ❖ متطلبات غير وظيفية : مثل الاعتمادية والأمن وسهولة الاستعمال وقابلية الصيانة...

لكي نحدد متطلبات المستخدم بالكامل ، هناك بعض البنود التي يجب أن يحددها محلل النظم ، وهي :

1- ملخص المسألة : تعريف لنشاطات المستخدم والمشاكل التي تحتاج إلى حل. مثلا قد يوجد لدينا متجر لبيع الكتب نقدا أو على الحساب. ويشعر صاحب المتجر بالحاجة إلى تطوير نظام سجلات الزبائن لزيادة الدخل.

2- الأهداف : غاية المنظومة الجديدة. مثلا يجب على المنظومة الجديدة أن تسرع معاملة معينة بنسبة 50٪. وكمثال آخر قد يطلب من المنظومة الجديدة توفير تقارير فورية للإدارة من خلال شبكة.

3- الحدود : أي نطاق العمل المطلوب. مثلا قد تحوي منظومة مبيعات على معلومات عن الزبون والمبيعات ولكن دون أن تشتمل على بيانات المخزون.

4- البيئة : الكينونات الخارجية التي تحيط بالنظام. ، مثلا قد يكون نظام المخزون بيئة لنظام المبيعات وأيضا يعتبر الزبون في منظومة المبيعات من البيئة الخارجية (أي كينونة خارجية)..

5- القيود : وهي الضوابط أو الكوابح التي تفرض على المنظومة المقترحة مثل الوقت المتوفر والميزانية المرصودة. فمثلا قد يكون قيد الميزانية 8000 دينارا مع ضمان 6 أشهر، وقيد الزمن هو سنتان من تاريخ العقد. وقد تكون القيود أيضا العتاد المستخدم ونوع الشبكة المستخدمة.

نشاطات مرحلة التحليل

1- إيجاد الحقائق و جمع المتطلبات: الخطوة الأولى في مرحلة التحليل هي فهم كيف يعمل النظام الحالي والمشاكل التى تعترضه ومن تمّ جمع وأستنباط المتطلبات للمنظومة المطلوب إعدادها.

ولتحقيق هذا الغرض المذكور يستطيع محلل النظم أن يستخدم الطرق التالية:

- ❖ البحث العام
- ❖ الاستبيانات
- ❖ المقابلات

- ❖ العرض التجريبي
- ❖ الملاحظة
- ❖ إثارة الأفكار الجديدة
- ❖ الاطلاع على مشاريع مشابهة
- ❖ الاطلاع على عينات من التقارير والنماذج المستخدمة
- ❖ اقامة ورش عمل "التصميم المشترك للتطبيق" JAD

ولدراسة معنى مصادر المتطلبات المذكورة أعلاه انظر الجدول التالي:

مصـــادر جمع المتطلبات :

المصـــــدر	التعريـــــــف
البحث العام	عمل بحث واستقصاء حول النظام باستخدام تسهيلات المكتبة والانترنت.
الاستبيان	أسئلة مطبوعة على ورق ذات علاقة بالنظام تعطى للمستخدم مع تعليمات واضحة لطريقة الاجابة عليها.
الملاحظة	عملية مشاهدة وتسجيل الوقائع والأحداث على أرض الواقع لفهم طريقة عمل النظام الحالي.
المقابلات المباشرة	الاجتماع مع الزبائن والمستخدمين وجها لوجه للاستفهام عن دور كل منهم في النظام الحالي والمشاكل التي يواجهونه والمتطلبات المرغوبة.

المقابلات غير المباشرة	الحصول على المعلومات عن طريق الاتصالات غير المباشرة مثل المكالمات الهاتفية ، والبريد الالكتروني والمحادثة عن طريق الانترنت والمؤتمرات عن بعد.
العينات Sampling	جمع عينات من النماذج والتقارير والوثائق التي يمكن أن تساهم في فهم المدخلات والمخرجات المطلوبة.
ملاحظات النموذج التجريبي prototype	منظومة تجريبية يتم عرضها على المستخدم لتلقي بعض الملاحظات التي تساعد في فهم احتياجاته.
إثارة الأفكار الجديدة Brainstorming	توليد أفكار جديدة عن طريق المناقشة الجماعية لأعضاء الفريق. يحصل رئيس الجلسة على أفكار جديدة بطرحه أسئلة ذات علاقة بالمشكلة.
ورشة العمل المشترك JAD	Joint Application Development التصميم المشترك للتطبيقات. ويعني تكوين فريق العمل المشترك من المستخدم ومعد المنظومة في شكل جلسات (ورش عمل) في مراحل التحليل والتصميم والاختبار.

المقابلـــة Interview

تعتبر المقابلة الشخصية وسيلة فنية لاكتشاف الحقائق وتحديد المتطلبات .وهي عبارة عن اجتماع تقابلي بين محلل النظم والزبون والمستخدم.

والغرض من هذا الاجتماع هو الحصول على معلومات حول النظام الحالي ومتطلبات المنظومة الجديدة .ويقوم محلل النظم عادة باجراء نوعين من المقابلات:

❖ النوع الأول يتم أثناء إعداد دراسة الجدوى في مرحلة التخطيط.هنا تجري مناقشة أولية مع الإدراة العليا والمتوسطة لتحديد الحدود والأهداف للمنظومة الجديدة. المرشحون للمقابلات يجب اختيارهم من المخطط الهيكلي للمؤسسة.

❖ النوع الثاني من المقابلات يجري مع الأشخاص من جميع المستويات في المؤسسة لفهم بالتفصيل خطوات أداء الوظائف والعمليات للنظام القائم. ويكون هذا في مرحلة التحليل.

ويمكن أن نقسم الأسئلة التي يتم توجيهها في المقابلة إلى 3 اصناف:

أ. أسئلـــة بإجابة قصيرة Closed-Ended Questions

تسمى أيضا أسئلة ذات نهاية مغلقة.

وتستخدم هذه الأسئلة عندما يريد محلل النظم أجابة محددة.

أمثلـة :

❖ كم عدد الموظفين في نظام الرواتب في المؤسسة؟

❖ كم عدد الطلبيات التي تنجزونها في اليوم الواحد؟

❖ كم من الوقت تستغرق عملية انجاز طلبية؟

ب. أسئلـــة بإجابات مفصلة Open-Ended Questions

تسمى أيضا أسئلة ذات نهاية مفتوحة.

وهي تعطي اجابات مفصلة وعفوية لأسئلة يسألها محلل النظم.

أمثلة :

❖ كيف تقوم باجراء "سحب صك" من المصرف؟ماهي الخطوات المتبعة بالتفصيل.

❖ كيف تحسب ضريبة الدخل ؟ لماذا تحسب بهذه الطريقة؟

❖ ما هي الطريقة التي تستخدمها لإعارة كتاب من المكتبة؟أي ماهو السيناريو المتبع لاداء هذه الوظيفة.

ت. أسئلـــة متدرجـــة الإجابات Scaled Questions

هذه الأسئلة ذات اختيارات محددة الاجابة باستخدام مقياس معين مثل { من 1 إلى 10 } أو { ضعيف ، متوسط ، جيد }.

أمثلة :

❖ كيف تقيّم عمل النظام؟ { ضعيف ، متوسط ، جيد }

- ❖ كيف تقيّم عمل موظفي القسم { من 1 الى 10 }
- ❖ هل يتعطل عتاد الحاسوب؟ { أبدا ، أحيانا ، دائما }.

إرشـــادات حـــول المقابلات

كمحلـل نظـم، يجـب عليـك اتبـاع الارشـادات التاليـة عنـد اجـراء المقابلات:

- ❖ ضع جدول زمني للمقابلات
- ❖ قدّم نفسك
- ❖ بين أهداف المقابلة
- ❖ كن صبورا مع التحضير والاستماع الجيد.
- ❖ لخص نتيجة المقابلة وراجع ما كتبته مع الشخص الذي قابلته.
- ❖ يمكنك استعمال اشرطة التسجيل في المقابلة ولكن بعد أخذ الإذن من الشخص الذي ستقابله.
- ❖ اكتب ملخص المقابلة ثم راجعها واحفظها وقيمها.

الملاحظـــة Observation

تعتبر الملاحظة وسيلة أخرى يستخدمها محلل النظم لفهم الاجراءات العملية للنظام الحالي بأن يتواجد في الموقع ويشاهد النظام واقعيا في حالة عمل .

وهنا يجب عليه أن يقارن بين ما استنتجه من المقابلات الشخصية مع الحقائق التى تمت ملاحظتهاعلى أرض الواقع.

ولكي يقوم بأخذ ملاحظات ملائمة ، عليه أن يتبع الارشادات التالية:

- ❖ جهز قائمة بالمهام التي تنوي ملاحظتها.
- ❖ جهز قائمة بالأسئلة ذات العلاقة بالمهام تحت الملاحظة .
- ❖ يمكن جمع النماذج المستخدمة أثناء أداء وظائف النظام من قبل الموظف المختص.

امثلـــــة:

- ❖ ماذا يحدث عندما يطلب الزبون سلعة ولكنها غير متوفرة في المخزن؟
- ❖ لاحظ ما يحدث عند سحب صك مصرفي واسأل ماذا يحدث إذا كان التوقيع غير مطابق.
- ❖ افحص عينة من النماذج والتقارير وجهز أسئلة عن غاية أستخدام كل منها.
- ❖ لاحظ انواع المهام التي يقوم بها كل موظف ودوره في المؤسسة.

الاستبيـــان Questionnaire

يعتبر طريقة أخرى لإيجاد الحقائق للحصول على بيانات من عدد كبير من الناس .

ويعطي الاستبيان عادة على شكل وثيقة(مجموعة أسئلة على ورق) تحتوي على عدد من الأسئلة بخصوص النظام الحالي ومقترحات لتحسينه.

كأمثلة على البيانات التي يتم تجميعها من الاستبيان ما يلي:

- ❖ حجم المعاملات المنجزة
- ❖ نوع المعاملات المنجزة.
- ❖ المدة التي تستغرقها الخدمة لمعاملة معينة .
- ❖ الصعوبات في انجاز المعاملات
- ❖ أي مقترحات لتحسين الخدمات

كأمثلة على الأسئلة في استبيان مايلي:

- ❖ كم من الوقت تستغرق في المتوسط عملية (سحب صك)؟
- ❖ هل لديك أي مقترحات لتحسين النظام الحالي؟
- ❖ كم من الوقت تستغرق عملية إعداد تقرير يدوي (الأصناف غير المتوفرة بالمخزون).

إرشـــادات بخصوص إعداد الاستبيان

- ❖ اجعل الأسئلة في الاستبيان مختصرة وسهلة مستخدما مصطلحات بسيطة.

- ❖ اكتب الأسئلة بحيث يكون لها اجابات قصيرة(أسئلة ذات نهايات مغلقة).
- ❖ رتب الأسئلة مبتدئا بالأسهل ثم الأكثر تعقيدا.
- ❖ في نهاية الاستبيان اترك فراغا للملاحظات والاقتراحات.
- ❖ ابدأ بمجموعة صغيرة من الناس قبل أن تعطي الاستبيان لمجموعة كبيرة.

البحث Research

هذه طريقة أخرى ذات فاعلية في جمع الحقائق.
باستخدام البحث يمكن لمحلل النظم أن يطلع على المصطلحات والاجراءات المستعملة في النظام تحت الدراسة.ويستحسن أداء هذه الطريقة قبل اجراء المقابلة الشخصية والملاحظة.

في عملية البحث يمكن لمحلل النظم أن يراجع ما يلي:

- الكتب
- المجلات
- الدوريات
- الانترنت

ويمكن لمحلل النظم الاتصال بالشركات الأخرى التي أنجزت نفس التطبيق للاستفادة من تجربتها والفائدة من البحث بالطبع اطلاع محلل

النظم بالمصطلحات والوظائف المستخدمة في النظام المراد تنفيده والاسفادة في اجراء المقابلة الشخصية.

تجميع عينات من التقارير والنماذج Sampling

يجب على محلل النظم أن يحصل على عينات من النماذج والتقارير المستخدمة حاليا.

هذه العينات من النماذج والتقارير يمكن الحصول عليها أثناء المقابلات مع الذين ينفذون العمليات أو أثناء الملاحظة.

إن تحليل هذه العينات يعطيه صورة واضحة عن كيفية عمل النظام الحالي.

لاحظ أنه يمكن أخذ عينات فارغة أو سبق ملؤها، ولكن في الحالة الأخيرة يجب أن تكون نماذج ملغية.

التصميم المشترك للتطبيق

(جاد (JAD=Joint Application Design

- تعتمد هذه الطريقة على فكرة العمل الجماعي.

هنا لدينا مجموعة من المستخدمين والمدراء وأعضاء فريق العمل يعملون معا لتحديد متطلبات المنظومة المقترحة والمشاركة معا خصوصا في التحليل والتصميم والاختبار.

وفكرة هذه الطريقة أن الزبون والمستخدم يعتبر طرفا رئيسيا في فريق عمل المنظومة وفي أغلب المراحل، ويكون ذلك في شكل ورش عمل في شكل جلسات مشتركة بين محلل النظم والمستخدم.

ويتكون فريق جاد من :-

- ❖ مدير المشروع
- ❖ المستخدمون والزبائن
- ❖ موظفي تقنية المعلومات
- ❖ محلل النظم
- ❖ مسجل المشروع

أهداف (جاد) الرئيسية هي:

- ❖ دراسة وتحليل النظام الحالي.
- ❖ إيجاد حلول بديلة
- ❖ تحديد متطلبات المنظومة الجديدة.
- ❖ تصميم المنظومة
- ❖ التقييم والاختبار

ملاحظات :

- ❖ يمكن أن يستخدم فريق (جاد) طريقة (إثارة الأفكار) لتعريف وتحسين المتطلبات.

- لكونه عضوا في فريق العمل في فريق (جاد) لن يجد المستخدم مشاكل في استخدام المنظومة ولن يحتاج تدريب عند إنتهاء إعداد المنظومة أي أن التدريب يبدأ في مراحل مبكرة.
- قبول المنظومة يتم تلقائيا لأن الزبون عضو مشارك في معظم مراحل الإعدادوبالتالي يكون مطلعا عليها طوال فترة أعداد المنظومة.

ملاحظات عامة حول جمع الحقائق والمتطلبات:

- المقابلات الشخصية يجب تحديدها بتاريخ ومكان وفترة معينة.
- أهداف المقابلة يجب أن تتحدد مسبقا.
- يجب أن يكون الاستبيان قصيرا.
- في الاستبيان استعمل نعم- لا - لا أعرف كإجابة، أو ضع اجابات اختيارية.
- عند الملاحظة، نسق زيارتك مع الإدارة والموظفين المتواجدين
- خذ ملاحظات قصيرة عند اجرائك للملاحظة.
- راجع ما كتبته مع المستخدم.

2- تحليل المتطلبات Requirements Analysis

بعد تجميع المتطلبات بأستخدام الطرق التي تمّ عرضها كالمقابلة والملاحظة يجب تنظيمها للحصول على مزيد من الفهم للنظام القائم والمقترح لتصبح أساس للمرحلة اللاحقة الا وهي مرحلة التصميم.

يستخدم المحلل لتحليل المتطلبات مجموعة أدوات مثل:

- مخطط انسياب البيانات Data Flow Diagram
- قاموس البيانات Data Dictionary
- جدول القرار Decision Table
- شجرة القرار Decision Tree
- الانجليزية المركبة Structured English
- مخطط الكائنات العلائقية Entity Relationship Diagram (ERD)

ملاحظة:

- مخطط الكائنات العلائقية يستخدم لتمثيل أو نمذجة البيانات الخاصة بالمنظومة وهو أساس تصميم قاعدة البيانات لن يتم التطرق له في هذا الفصل وسيتم التطرق له في الفصول اللاحقة.
- يجب على محلل النظم اعداد وثيقة في نهاية مرحلة التحليل.
- تسمى هذه الوثيقة بوثيقة (توصيف المتطلبات) المطلوبة .

وهي تصف الوظائف المتعلقة بالمنظومة وخصائص هذه الوظائف.

وقد تحتوي وثيقة تحديد المتطلبات البنود التالية:

1. ملخص المشكلة
2. مخطط انسياب البيانات
3. قاموس البيانات
4. وظائف المنظومة
5. متطلبات الأداء
6. البيئة
7. القيود
8. حدود المنظومة
9. معيار القبول
10. لغة البرمجة المستخدمة
11. شجرة القرار decision table
12. جدول القرار decision tree
13. الانجليزية المركبة structured English
14. مخطط العلاقات الكائنية (ERD)

ملاحظات عن وثيقة تحديد المواصفات (وثيقة المتطلبات):

- تصف مشاكل وليس حلولا.
- هى منتوج وليس عملية معالجة.

- إنها وثيقة بين الزبون والمحلل (المنتج)و سوف تستخدم فيما بعد في التصميم.
- تقوم بتحويل الاحتياجات إلى متطلبات.
- يجب مراجعتها من قبل المستخدم ومعد المنظومة.
- تبين ما هو المتوقع من المنظومة وليس كيف تعمل.

ملاحظة:

نستخدم عادة المصطلح "احتياجات المستخدم" والمصطلح "متطلبات المستخدم" كما لو كانا بنفس المعنى. ولكن في الواقع هناك فرق حيث المصطلح "متطلبات" أدق وأكثر قابلية للاختبار.

لتوضيح الفروق بين الاحتياجات والمتطلبات ندرس الأمثلة التالية:

REQUIREMENT المتطلبات	NEED الاحتياج
1000معاملة مباشرة On-Line خلال 24 ساعة.	النظام المقترح يجب أن يكون مباشرا على الخط On-Line وينفذ العديد من المعاملات يوميا.
الملف يحمل إلى غاية 100000 سجل زبون.	الملف يجب أن يحمل أسماء كل زبائن الشركة.

لاحظ في الامثلة السابقة أن المتطلبات أكثر دقة وقابلة للقياس بمعنى انها تستخدم أرقام من السهل أختبارها ومعايرتها.

المتطلبات يجب أن تمتلك الخصائص التالية:

- كاملة
- دقيقة
- موجزة
- واضحة
- متناسقة (عدم التضارب)
- قابلة للقياس

ملاحظة :

سيتم مناقشة أدوات هذه المرحلة في هذا الكتاب في الفصل اللاحق .

3.5 مرحلة التصميم Design phase

تقوم عملية التصميم بترجمة المتطلبات إلى تمثيل الحل.

التصميم يركز على:

1. تصميم معماري للنظام **Architectural Design**

يتم تجزئة المنظومة باستخدام *المخطط الهيكلي*.

2. تصميم واجهة المستخدم

Graphical User Interace Design

ويتم هنا تصميم واجهة المستخدم للمدخلات والمخرجات وهي عادة ما توصف باستخدام الشاشات(النماذج) والتقاريرويتم عادة بأستخدام لغة برمجة مرئية مثل بيسك المرئي أو من خلال لغة قاعدة البيانات مثل SQL.

3. تصميم البيانات **Data Design**: وصف للملفات أوقاعدة البيانات.

وتستخدم جداول قاعدة البيانات وجداول قاموس البيانات لاعداد هذا التصميم.

4. تصميم الخوارزميات **Algorithmic Design**:

وتعني اعداد خوارزمية لكل جزء برمجي في المخطط الهيكلي باستخدام أداة شبه الشفرة pseudocode .

وفي نهاية هذه المرحلة يتم اعداد وثيقة مواصفات التصميم التي قد تحتوي على ما يلي:

- ❖ تصميم الشاشات والتقارير.
- ❖ نمذجة البيانات باستعمال مخطط الكائنات العلائقية / قاموس البيانات / الجداول القياسية.
- ❖ التصميم المعماري للبرمجيات باستخدام المخطط الهيكلي .
- ❖ خوارزميات كل جزء برمجي باستعمال شبيهة الشفرة أو المخطط الانسيابي الهيكلي
- ❖ تحديد أجزاء المكونات المادية hardware التي تلزم للنظام المقترح وكذلك نوع الشبكة .
- ❖ تحديد البرمجيات المستخدمة لاعداد المنظومة مثل المترجمات ونظام التشغيل.

3.6 مرحلـــة التنفيذ Impementation Phase

ان الهدف الرئيسي لمرحلة التنفيذ انتاج شفرة مصدر source code التي عند تنفيذه على جهاز الحاسوب نحصل على منظومة تعمل بكفاءة (بدون أخطاء) وتسمى هذه المرحلة أيضا بمرحلة البرمجة أو مرحلة التشفير.

وتتعلق المرحلة التنفيذية بالنشاطات التالية:

- ❖ ترجمة مواصفات التصميم لكل جزء برمجي إلى شفرة مصدر.
- ❖ تصحيح الأخطاء.
- ❖ اجراء اختبار للوحدات (أي للأجزاء البرمجية)
- ❖ كتابة كتيب التشغيل

ملاحظة : يعتمد عدد المبرمجين في هذه المرحلة على حجم المشروع وهيكلية الفريق الذي تم تشكيله عند تخطيط المشروع.

وعند مرحلة البرمجة يجب تحقيق الآتي:

- ❖ استعمل لغة البرمجة التي تناسب التطبيق المطلوب.
- ❖ اجعل قابلية قراءة المخرجات أفضل ما يمكن باستعمال الواجهة الرسومية.
- ❖ اجعل قابلية قراءة شفرة المصدر أفضل ما يمكن باستعمال أسماء ذات معنى وجمل تعليقية.

- ❖ اجعل وقت اعداد المنظومة أقل ما يمكن باستخدام لغة برمجة ذات كفاءة عالية.
- ❖ اجعل عدد جمل شفرة المصدر أقل ما يمكن.
- ❖ اجعل الذاكرة المطلوبة أقل ما يمكن.

ويجب أن يحقق الجزء البرمجي ما يلي:

- أن يكون له مدخل واحد ومخرج واحد.
- لا يوجد به جمل GOTO إلا للضرورة.
- أن يحتوي على أقل من 50 جملة (أي حوالي صفحة).
- أن يستعمل التنسيق والأسطر الخالية لجعل الجزء البرمجي هيكليا وواضحا.
- ألا يتداخل بعمق كبير(مثلا لا يزيد عن 5 جمل شرطية أو 5 حلقات طالما متداخلة).
- أن يحتوي على توثيق كافي.

ملاحظة: في هذه المرحلة يجب أن يتم أعداد كتيب المشغل لمعرفة كيفية تشغيل المنظومة ويتم أستعماله فى الاختبار والتدريب من قبل المستخدم.

3.7 مرحلـــة الاختبار Testing Phase

الهدف الرئيسي من هذا الاختبار هو تحديد امكانية الحصول على النتائج المطلوبة عند تشغيل المنظومة وبدون أخطاء.

وهذا يعني أن اختبار المنظومة الجديدة يجب أن يحقق:

- ❖ التخلص من الأخطاء اللغوية
- ❖ التخلص من الأخطاء المنطقية
- ❖ التخلص من الأخطاءالتنفيذية
- ❖ تحديد أخطاء مدخلات المستخدم
- ❖ تقييم سرعة أداء المنظومة
- ❖ تقييم أمن المنظومة
- ❖اكتشاف أي وظيفة مفقودة.
- ❖ تحقيق متطلبات المستخدم
- ❖تقييم توثيق المنظومة

ولاختبار منظومة جديدة عادة ما نستعمل عينة من البيانات ونقارن المخرجات (النتيجة) مع النتيجة المتحصل عليها من نفس العينة يدويا. ويجب أن تحتوي هذه العينة على جميع الحالات المحتملة للتأكد من عدم وجود أخطاء منطقية .

وكما ذكرنا سابقا فإن كل جزء برمجي (برنامج فرعي) من المنظومة يتم اختباره منفصلا قبل اختبار المنظومة ككل. يشمل الاختبار عدة أنواع ولكن من أهمها الاختبارات الاتية:

- ❖ اختبار التكامل Integration test
- ❖ اختبارالنظام System Test

- ❖ اختبار القبول Acceptance Test

في اختبار التكامل يتم ربط واختبار الأجزاء البرمجية (التى تم أختبارها منفردة مسبقا) كنظام متكامل(وحدة واحدة) للتأكد من أن المتطلبات كما عرّفها المستخدم- قد تم تحقيقها.

اختبار النظام يتحقق من جميع المتطلبات باستخدام عتاد وبيانات حقيقية من واقع العمل . ويكون ذلك بعد أختبار التكامل وقبل أختبار القبول.

اختبار القبول يتم تنفيذه من قبل المستخدم لاثبات أن المتطلبات المدونة فى وثيقة المتطلبات قد تم أنجازها وأن الزبون موافق على المنظومة وأنها جاهزة للعمل.وبالطبع هناك أكثر من 12 نوع من الاختبارات التي لن يتم التطرق لها في هذا الكتاب ولكن سيتم التطرق لها في كتاب "هندسة البرمجيات" وهو من تأليف المؤلف لهذا الكتاب مع العلم بأن كتاب هندسة البرمجيات موجود حاليا في المكتبات.

3.8 مرحلة الصيانة Maintenance Phase

الصيانة هي عملية جعل المنظومة بعد استلامها من قبل الزبون وفي فترة عملها تعمل بطريقة صحيحة في مواجهة العوامل التالية التي قد تؤثر على عملها:

- ❖ التغيرات بسبب أخطاء .
- ❖ استخدام تقنيات جديدة
- ❖ مشاكل قد تحدث في المنظومة بسبب رداءة في التصميم(أعادة هندسة المنظومة).
- ❖ الرغبة في إضافة وظائف جديدة وتحسين قدرات المنظومة.

و تبدأ الصيانة عندما يتم تسليم المنظومة وتركيبها(تحميلها) في موقع الزبون والبدء في تشغيلها.

لهذا نقترح عمل عقد صيانة مع معد المنظومة بعد أستلامها وتشغيلها .

النشاطات التالية يتم انجازها في هذه المرحلة:

1. **تصحيح الاخطاء**

يتعلق بتصحيح الأخطاء التي تظهر أثناء عملية تشغيل المنظومة والتى لم يتم أكتشافها فى مرحلة الاختبار.

2. **التكيف**

نظرا للتغيرات السريعة في التقنية (برمجيات ومعدات) قد تحتاج المنظومة الى التكيف مع بيئة جديدة مثل استخدام نظام تشغيل حديث أو معدات حاسوب أو برمجيات حديثة أو أستخدام شبكة.

3. **التحسين**

ويشمل إما تطوير امكانيات المنظومة أو توفير وظائف أكثر للمنظومة عند اكتشاف متطلبات جديدة. فمثلا قد تطلب الادارة أنواع جديدة من التقارير أو أضافة عمليات جديدة في أقسام المؤسسة تستدعي أضافتها للمنظومة.

4. **إعادة الهندسة :**

نقوم هنا بإعادة تصميم النظام وبرمجته - إذا لزم الأمر- لمنع أي مشاكل متوقعة من التصميم السيئ أو تقليص مدة الصيانة.

3.9 مراجعة البرمجيات Software Review

تعتبر المراجعة نشاطا يمارس في نهاية كل مرحلة كمصفي لعملية إعداد البرمجيات على شكل جلسة يحضرها أعضاء فريق إعداد المنظومة وغالبا ما يشارك فيها المستخدم والزبون.

أهداف المراجعة :

- ❖ اكتشاف الأخطاء في مواصفات أي مرحلة.
- ❖ التأكد من أن متطلبات المنظومة قد تحققت .
- ❖ التأكد من أن المنظومة قد تم اعدادها (انتاجها) طبقا لمعايير الجودة التي تم ذكرها في وثيقة المتطلبات .
- ❖ المصادقة على المواصفات بغية الانتقال الى المرحلة اللاحقة.

وهذا يعني :

- ❖ مراجعة وثيقة توصيف المتطلبات في نهاية مرحلة التحليل.
- ❖ مراجعة وثيقة توصيف التصميم في نهاية مرحلة التصميم.
- ❖ مراجعة شفرة المصدر في مرحلة البرمجة.

جلسة المراجعة يحضرها كل من:

- ❖ موظفي تأكيد الجودة
- ❖ منتجو المنتوج البرمجي
- ❖ الإدارة
- ❖ ممثل المستخدم : قد يكون مكتب استشاري ينوب أو يشارك المستخدم
- ❖ المستخدم

في نهاية المراجعة هناك 3 احتمالات:

- ❖ قبول العمل بدون تعديل.
- ❖ رفض العمل بسبب أخطاء فادحة.
- ❖ قبول العمل بشرط تعديل الأخطاء البسيطة.

يتم اعداد تقرير مختصر في نهاية المراجعة ليتم الاستفادة منه في المراحل اللاحقة.

3.10 التـوثيق Documentation

يعتبر التوثيق جزءا مهما من أجزاء إعداد واستخدام المنظومةالبرمجية.

ولكن هذا التوثيق عادة ما يهمل من قبل كل من معدي المنظومات والمستخدمين.

لهذا نجد أن معظم معدي البرمجيات يفتقدون الأدوات التي تساعدهم في إعداد توثيق كامل وصحيح.

وكقاعدة، يجب تخصيص 20 ٪ من مجهود إعداد البرمجيات لتوثيق المنظومة الجديدة.

تعريف: يمكننا تعريف التوثيق بأنه مجموعة أوصاف وشروح المنتوج البرمجي وما يتعلق به.

وقد يشمل التوثيق ما يلي:

- ❖ نص أو سرد
- ❖ مخططات
- ❖ جداول
- ❖ الصوت
- ❖ قصاصات فيديو
- ❖ صور متحركة
- ❖ تعليقات داخل شفرة البرنامج

ويمكن أن تكون الوثيقة على شكل ورقة Hardcopy أو تكون مخزنة في الحاسوب Softcopy.

وظائف التوثيق:

يؤدي التوثيق الوظائف التالية:

- ❖ مرجع تاريخي
- ❖ مرجع إرشادي مثل كتيب التشغيل.
- ❖ .التحكم في الكم والجودة
- ❖ التواصل بين مراحل أعداد المنظومة
- ❖ التواصل بين المهام
- ❖ اتفاق أو عقد بين المستخدم أو الزبون ومعد المنظومة

أنواع التوثيق:

ويمكن تقسيم التوثيق إلى صنفين:

1. توثيق للمعدّين : Developers Documentation

وهو توثيق خاص للمبرمجين ومحللي النظم والمصممين ...

ويوجد هذا النوع من التوثيق في : -

- ❖ البرنامج نفسه (مثلا جمل تعليق في لغة C أو ملاحظات في لغة البيسك).

- ❖ دليل منفصل عن البرنامج مثل كتيب الصيانة او وثيقة المتطلبات.

توثيق المعد يستخدم بصورة عامة من قبل:

- ❖ الإدارة{ لغرض المراجعة}
- ❖ القائمون على صيانة البرنامج
- ❖ فريق التفتيش
- ❖ زملاء العمل المراجعة غير الرسمية من موظفي التحقق والمصادقة

ملاحظة: لاحظ أن هذا النوع من التوثيق عادة ما يصف تركيبة البرنامج باستخدام المخططات (مثلا مخطط انسياب البيانات وشبيهة الشفرة والمخططات المركبة والبيانات الاختبارية وقاموس البيانات...)

2. توثيق المستخدم User Documentation

هذا النوع من التوثيق يقرأه مستخدموا المنظومة. وهو عادة ما يكون على شكل دليل المستخدم.

محتويات دليل المستخدم قد تشمل الآتي:

- ❖ .الدخول والخروج من المنظومة
- ❖ وصف لأهم سمات المنظومة.

❖ قائمة بوظائف المنظومة

❖ متطلبات العتاد

❖ ملخص لشكل التقارير والشاشات

❖ :شكلية التشغيل

- الأيقونات
- القوائم المندفعة
- القوائم القياسية
- مربعات الحوار

❖ عينات تبين المدخلات والمخرجات والأفعال الواجب عملها في حالة وجود أخطاء(مثال عملي لاضافة صنف ،استفسار عن صنف في منظومة المخازن).

❖ مزايا متقدمة :

- عرض مرئي
- الصوت

❖ مصطلحات المنظومة.

❖ قائمة برسائل الأخطاء

❖ طلب المساعدة help

❖ الأمن security .

بديهيا ، يجب أن يكون دليل المستخدم بسيطا وواضحا حتى يتمكن أي مستخدم من فهمه. لهذا يجب أن تتم كتابته من قبل كتاب تقنيين(Technical Writers)

ما يجب أن يسجله التوثيق :

❖ كيف تعمل المنظومة الجديدة (مثلا دليل المستخدم).

❖ كيف تم تصميم المنظومة الجديدة (مثلا وثيقة توصيف التصميم).

❖ كيف يمكن صيانة المنظومة الجديدة (مثل وثيقة الصيانة).

الوثائق المعدة في نهاية كل مرحلة

اسم الوثيقة	المرحلة
▪ طلب المستخدم ▪ دراسة الجدوى ▪ -خطة المشروع ▪ مقترح المشروع	التخطيط
▪ -وثيقة المتطلبات	التحليل
▪ وثيقة توصيف التصميم	التصميم
▪ شفرة البرنامج ▪ تقرير نتيجة اختبار البرنامج الجزئي	التنفيذ

▪ تقرير اختبار التكامل ▪ تقرير اختبار النظام ▪ تقرير اختبار المستخدم ▪ دليل المستخدم ▪ المنظومة النهائية	الاختبار
▪ دليل الصيانة ▪ دليل الإجراءات	الصيانة

يبين الجدول السابق الوثائق التي يتم اعدادها في نهاية كل مرحلة من مراحل دورة حياة أعداد المنظومة .

إليك الآن بعض الأمثلة على توثيق المعد:

. **وثيقة المتطلبات:** وهي وثيقة يتم إعدادها في نهاية مرحلة التحليل. هذه الوثيقة يجب أن تكون صحيحة ودقيقة وكاملة ومتناسقة وقابلة للقياس والاختبار.

يبين الشكل التالي البنود التي تشملها عادة هذه الوثيقة.

أ. **مقدمة**

1. **وصف عام**
   - تعريف المسألة
   - الأهداف
   - البينيات
   - حدود المنظومة
   - القيود
2. **الوصف الوظيفي:**
   - قائمة الوظائف
   - وصف كل وظيفة

ب. **وصف البيانات :**

- مخطط الكائنات العلائقية
- قاموس البيانات

ت. **وصف المعالجة والمنطق**

- مخطط انسياب البيانات
- مخطط استخدام الحالة
- الانجليزية المركبة
- شجرة القرار
- جدول القرار

ملاحظة : مخطط أستخدام الحالة يعتبر اداة تحليل كائنى لتحديد المتطلبات الوظيفية للنظام و ولكنه أيضا يعتبر اداة ممتازة فى التحليل الهيكلى.

ث. متطلبات الاداء:

- زمن الاستجابة
- الذاكرة

ج. معيار التحقيق والقبول:

- أنواع الاختبارات
- خصائص الجودة المطلوبة
- البنود القابلة للتسليم

ح. استراتيجية الحل :

- على الخط- On-line أم خارج الخطOff-Line
- مشاركة الوقت / موزع
- رسومات أم نص
- قاعدة بيانات أم ملفات
- شبكة /او عدم وجود شبكة

2. وثيقة خطة المشروع

سبق وأن ناقشنا محتويات هذه الوثيقة.

3. وثيقة توصيف التصميم

وهي تحتوي على البنود التالية:

أ. التصميم المبدئي:

- *واجهة المستخدم والتقارير*
- *المخطط الهيكلي*
- *تصميم مفهوم البيانات ERD*

ب. تصميم مفصل:

- *تصميم قاعدة البيانات (الجداول المعيارية) وقاموس البيانات*
- *تصميم البرامج الفرعية باستخدام شبيهة الشفرة أو المخططات الانسيابية الهيكلية.*

مزايـــــا التوثيق

1. للتأكد من استمرارية المشروع في حالة غياب أو توقف أو استبدال بعض أعضاء فريق إعداد المنظومة .
2. يمكن استعمال التوثيق كدليل مرجعي للمحللين والمبرمجين الذين ينفذون التغيير في المنظومة .
3. يستفاد من التوثيق في تقييم فاعلية التواصل بين المراحل والمهام الداخلية للمشروع.
4. تسهل عملية التفتيش والمراجعة.
5. يمكن أن يستخدم للمشاريع الجديدة والمشابهة.
6. تستطيع الإدارة اتخاذ قرارات حول المنظومة مبنية على التوثيق.
7. قد يستفيد المتدربون على المنظومة الجديدة من التوثيق (مثلا دليل المستخدم) كأداة سهلة لمعرفة كيفية استخدامها.

كتابة التوثيق

تزداد الحاجة كل يوم إلى منتوج برمجي ودود مع المستخدم، ومن ثم الحاجة إلى كتّاب فنيين Technical Writers للتوثيق.

الكتابة التقنية تضيف بعدا آخر لإعداد المنظومات.

يجب على الكتّاب التقنيين أن يكون لهم خلفية في المجالات التالية :

- ❖ علم الحاسوب
- ❖ نظم إدارة المعلومات
- ❖ القدرة على كتابة مستندات وتقارير وأدلة بصورة واضحة ودقيقة.

ملاحظة : عندما نكتب التوثيق، يجب أن نأخذ في الاعتبار المعايير القياسية في الكتابة التقنية . وهذه المقاييس يجب أن تغطي المستخدمين والعمليات والبرمجة، ..الخ.

3.11 أسئلة للمراجعة

1. يوجد العديد من الأسماء للنموذج التقليدي لاعداد المنظومات . ما هي؟
2. لماذا من المهم القيام بدراسة الجدوى قبل بداية أي مشروع؟
3. ما هي النشاطات التي تنجز في:

- ❖ التحليل المبدئي
- ❖ التحليل المفصل

4. قارن بين الاحتياجات والمتطلبات.
5. ما هي الخصائص التي يجب أن تكون في المتطلبات؟
6. لماذا يقوم المستخدمون بتغيير متطلباتهم خلال مراحل المشروع؟
7. ما هو الغرض من التصميم؟
8. ما نوع الأنشطة المطلوب أداؤها في مرحلة التصميم؟
9. في نهاية مرحلة التصميم ، يتم اعداد وثيقة التصميم. ما الذي يجب أن تحتويه هذه الوثيقة؟
10. ما هو الغرض من تنفيذ البرمجيات؟
11. لماذا من المهم أن يكون البرنامج المصدري واضحا؟
12. ما هي أنشطة مرحلة التنفيذ؟
13. ما هي الجوانب الواجب مراعاتها عند كتابة البرنامج المصدري؟
14. لماذا يجب أن نختبر المنظومة الجديدة؟
15. عرف :

❖اختبار التكامل ❖اختبار القبول

16. لماذا تعتبر مرحلة الصيانة ذات أهمية كبيرة في عملية إعداد البرمجيات؟

17. ما هي الأنشطة التي تزاول في مرحلة الصيانة؟

18. ماذا يقصد بمصطلح (المراجعة) ؟

19. تتم عملية المراجعة في نهاية كل مرحلة. وهذا يعني أن هناك على الأقل ثلاثة مراجعات أثناء المشروع. اذكر ماهي؟

20. من يجب أن يتواجد في جلسة المراجعة؟

21. ما هي الاحتمالات الممكنة في نهاية كل مراجعة؟

22. ما هو (التوثيق)؟ ما هي أهميته في عملية إعداد البرمجيات؟

23. عادة ما يأخذ التوثيق شكل النص أو الجداول. ما هي الأشكال الأخرى التي قد يأخذها؟

24. اكتب ما تعرفه عن نوعين من التوثيق .

25. ما أنواع المعلومات التي يجب أن يسجلها التوثيق؟

26. ما هي الوثائق التي يتم إعدادها في نهاية كل مرحلة من النموذج التساقطي (النموذج التقليدي)؟

27. ما هي الفوائد التي نحصل عليها من التوثيق؟

28. من يجب أن يكتب التوثيق؟

4

Chapter four

تخـــطيط المشروع

Project Plann

1. مقدمة
2. أنشطة التخطيط
3. مخطط تجزئة العمل
4. الجدولة وتقدير مدة المشروع
5. أدوات الجدولة للمشروع
6. مخطط غانت Gantt Chart
7. جدول الخط الزمني للمشروع
8. تحليل الأخطار
9. أسئلة للمراجعة

- الأجهزة
- البرمجيات
- العنصر البشري

والقيود في هذه الحالة قد تكون كما يلي:

- التكلفة : الميزانية محددة بقيمة 10000 دينار.
- الوقت: يجب ألا يزيد زمن المشروع عن 5 أشهر.

4. تجهيز جدول زمني للمشروع .

مثال : قد يكون الجدول الزمني لمشروع مدته 10 أشهر على النحو التالي:

- التحليل شهران
- التصميم شهران
- البرمجة 3 أشهر
- الاختبار 3 أشهر

5. تعريف وظائف المنظومة التي سيجري إعدادها بحيث يتم انجاز المتطلبات ولو جزئيا.

مثال : وظائف منظومة المخزون يمكن أن تكون على النحو التالي:

- اضافة أصناف جديدة للمخزون
- الاستفسار عن صنف معين

- تحديث بيانات صنف معين.
- .تكوين تقارير

6. وضع المعايير التي ستسخدم في المشروع.

مثال : قد تكون بعض معايير منظومة مستشفى كما يلي:

- استخدام لغة بيسك المرئي VB.net و لغة الاستفسار SQL كأدوات برمجية.
- يجب أن تعمل المنظومة في بيئة شبكة network
- يجب ألا يزيد عدد الأسطر في البرنامج الفرعي عن 30.

7. تعريف خصائص الجودة المطلوب تحقيقها.

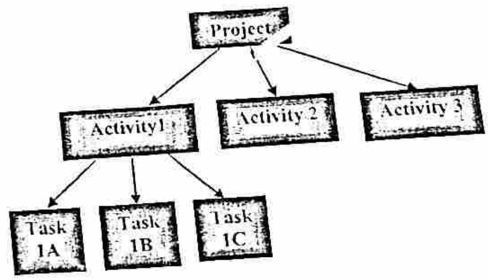
مثال : منظومة المصرف **banking system** يجب أن تتمتع بالاعتمادية ، والكفاءة ، وقابلية الصيانة وسهولة الاستعمال.

ونستخدم في مرحلة التخطيط مجموعة أدوات في شكل مخططات والتي منها :

4.3 مخطط تركيبة تجزئة العمل WBC
Work Break_down Structure

هذه الأداة WBS هي عبارة عن أداة تخطيط تبين حدود المشروع كمجموعة من المهام بتركيبة هرمية من فوق إلى تحت.

في الشكل (4.3.1) نجد شكلا عاما ل WBS حيث يقسم المشروع أولا إلى عدد من النشاطات، ثم يمكن أن يقسم كل نشاط إلى مجموعة مهام.



الشكل 4.3.1 مخطط (WBS (Process

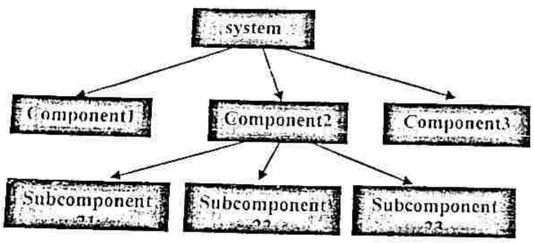
ويعتبر WBS أعلاه من نوع المعالجة Process WBS لأنه يصف عمليات المشروع يجب إنجازها.

ولكن يمكننا أن نعرّف نوعا آخر من WBS نسميه (نوع المنتوج Product WBS) كما يلي:

مخطط WBS هو أداة تخطيط هرمية على شكل مخطط يبين نطاق النظام كمجموعة من الأجزاء الفردية بطريقة من فوق إلى أعلى.

أي أن Process WBS يختص بالمشروع في شكل مراحل ونشاطات، وأن Product WBS يختص بالمنظومة ومكوناتها.

ويوضح الشكل (4.3.2) هذا التعريف حيث تقسم المنظومة System المطلوب إعدادها إلى مجموعة من المكونات Components ثم إلى مكونات فرعية (إذا لزم الأمر).



الشكل 4.3.2 WBS (Product)

مثال : ارسم مخطط من نوع WBS(المنتوج) لمنظومة مصرفية ذات ثلاثة مكونات فقط، هي :

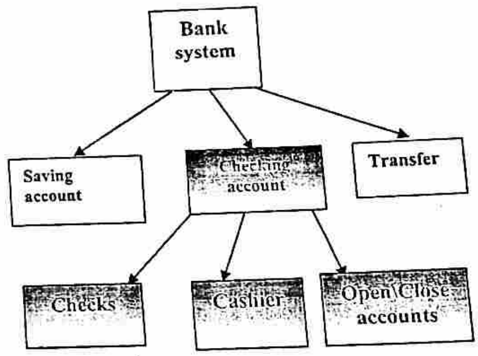
- حساب توفير
- حساب جاري
- تحويلات

حيث حدود المنظومة تخص جزئية الحساب الجاري فقط وهي ذات 3 مكونات جزئية هي:

- الصكوك

- الصراف
- فتح وقفل الحسابات

يبين الشكل 4.3.3 المخطط المطلوب. لاحظ تظليل المكونات التي ستسلم للمستخدم.



الشكل 4.3.3 مخطط WBS لنظام مصرفي

في هذا الشكل استخدمنا المصطلحات المصرفية التالية:

Bank system نظام مصرفي
Saving account حساب توفير
Checking account حساب جاري
Transfer حوالات

Checks صكوك
Cashier الخزينة
Open\close account فتح / قفل حساب

مميزات WBS :

1) أداة تخطيط قوية لتوضيح حدود المشروع.

2) أداة متابعة قوية لإعداد تقارير حول سير المشروع للإدارة والزبون.

3) يمكن استخدام مخطط WBS لتقدير التكلفة ومدة المشروع.

4) توفر آلية لقياس الأداء.

4.4 الجدولة وتقدير زمن المشروع

الجدولة وتقدير الوقت للمشاريع يعتبران الجزء الصعب في التخطيط. ويعتمد التقدير الزمني لأي مشروع على الفترة الزمنية التي يستغرقها انجاز كل مهمة من مهام هذا المشروع.

وعندما نقدر الزمن يجب أن نستعمل أيام العمل كوحدة قياس للزمن.
لاحظ أن ساعات الدوام قد تختلف من مكان إلى آخر، وفي هذه الحالة تستخدم الساعة كوحدة قياس للزمن.

إرشـــــادات بخصوص تقدير الوقت :

1) يجب التعرف على المهام التي يمكن أن تنجز على التوازي.
2) استعمل أقل قوى عاملة ممكنة .
3) خذ في الاعتبار أن بعض الناس قد يتركوا العمل أو يتعرضوا للمرض.
4) يتم التقدير كما لو سيكون كل شئ على ما يرام، ثم يتم زيادة 10٪ مثلا للمشاكل الغير متوقعة.

مثلا إذا كان الجزء البرمجي يستغرق يوما واحدا لانهائه ، وكان لدينا 30 جزءا برمجيا في المشروع، فإننا نقدر الزمن لكتابة كل الأجزاء بمدة 30 يوما. يضاف إلى ذلك 3 أيام (أي 10٪) للأحداث غير المتوقعة (مرض، إجازة، ..) فيصبح التقدير هو 33 يوما.

ملاحظة : ليكن في معلومك أن مواصفات المتطلبات والتصميم تتطلب ضعف الوقت اللازم لكتابة شفرة البرنامج.

4.5 أدوات الجدولـــة

أدوات الجدولة يجب أن تبين:

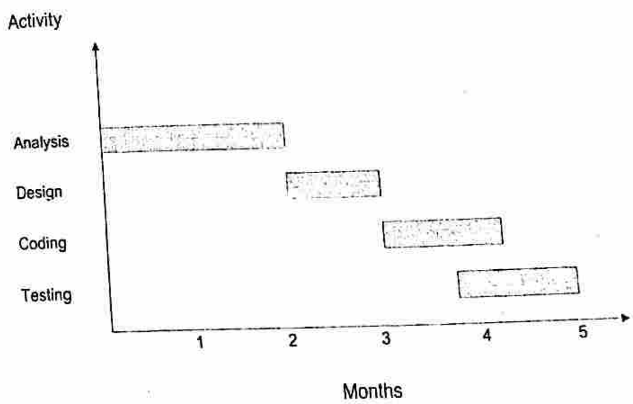
- ❖ المهام
- ❖ العلاقة بين المهام
- ❖ العنصر البشري المعين لكل مهمة

المخططـات إمـا أن تنجـز يـدويا ، أو بـأدوات خاصـة مثـل برنـامج " المشاريع project" الذي أعدته شركة مايكروسوفت.
هذه الأدوات تفترض أن المشروع يتكون من مجموعة من النشاطات أو المهام المعتمدة على بعض. فمنها مهام لا يمكن أن تبدأ قبل أن تنتهي مهام أخرى، ولكن بعض المهام يمكن أن تنجز بالتوازي مع نشاطات أخرى.
في عملية الجدولة ، يستخدم مديرو المشاريع أدوات مثل: مخططات غانت Gantt ، وجدول الخط الزمني، وطريقة CPM ، وطريقة PERT.
هذه الأدوات تستخدم في تخطيط ومتابعة المشاريع.

4.5.1 مخطط غـانـت Gantt Chart

مخطط غانت هو أداة تخطيط رسومية تستخدم لعرض مهام المشروع ومدتها الزمنية.
يمكن رسم مخطط غانت يدويا أو بواسطة برنامج رسم.

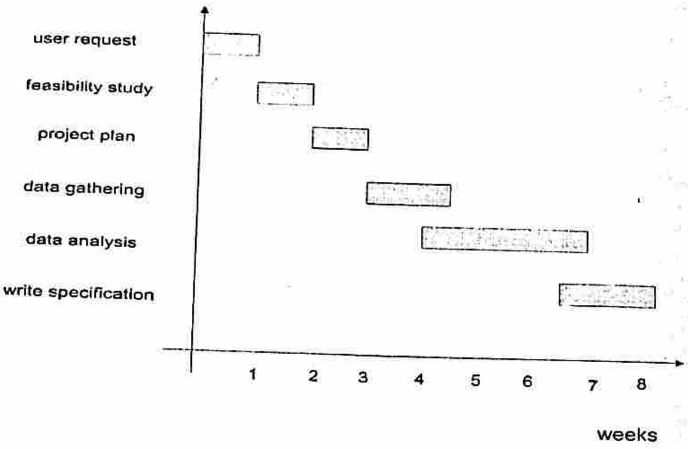
المحور الأفقي للمخطط يعبر عن الفترة الزمنية (مثلا بوحدات الأيام أو الأشهر أو الأسابيع) أما المحور العمودي فيمثل المراحل (أو المهام أو النشاطات)



الشكل 4.5.1 مخطط غانت لمشروع بالأشهر months

كما نرى في الشكل فإن مخطط غانت يبين المراحل المتداخلة مثل مرحلتي التنفيذ والاختبار.

في الشكل التالي نرى مثالا على مخطط يدوي لمشروع يتكون من 6 نشاطات حيث نعبر عن مدة كل نشاط بالاسابيع:



الشكل 4.5.2 مخطط غانت لمشروع بالأسابيعweeks

في الشكل السابق يمثل مخطط غانت مرحلتي : التخطيط والتحليل فقط.

لو مثلنا مهام النشاطات الأخرى لأصبح المخطط طويلا جدا.

لحل هذه المشكلة يمكن استخدام جدول الخط الزمني ليمثل مهام المشروع بالتفصيل.

مثلا يمكن توضيح الفترة الزمنية بالأيام بدلا من أسابيع وأشهر بكل سهولة في هذا الجدول..

4.6.1 جـــدول الخط الزمني Time-line Table

هذا عبارةعن أداة تخطيط ومتابعة ذات صفوف للمهمات، وأعمدة لتواريخ البداية والنهاية المتوقعة والفعلية، مع امكانية إضافةعمود لمن يقومون بكل مهمة.

(يمكن متابعة الجدول من الشمال الى اليمين).

المهمة task	البداية المتوقعة expected start	النهاية المتوقعة expected end	البداية الفعلية Actual start	النهاية الفعلية actual end

الشكل يعبر عن جدول الخط الزمني

نستطيع أيضا أن نضيف عمودين لهذا الجدول الأول للتكلفة المتوقعة والثاني للتكلفة الفعلية.

ملاحظة : في جدول الخط الزمني نقوم أولا بملء أعمدة البداية والنهاية المتوقعة قبل بداية المشروع.

أمّا البداية والنهاية الفعلية لكل مهمة يمكن أن تعرف أو تدوّن فقط أثناء وبعد عمل المشروع.

يقوم مدير المشروع بملأ البيانات الفعلية عند وقوعها.

وهذه العملية تساعد في إعادة التخطيط للمشروع ومعرفة أسباب التأخير.

افترض أن لدينا 4 مهام في نشاط تجميع البيانات كما يلي:

المهمة	البداية المتوقعة	النهاية المتوقعة	البداية الفعلية	النهاية الفعلية
Interview customer	1-1-2010	3-1-2011	1-1-2011	4-1-2011
Interview end-user	4-1-2010	10-1-2011	5-1-2011	11-1-2011
Observation الملاحظة	11-1-2010	17-1-2011	12-1-2011	18-1-2011
Questionaire الاستبيان	18-1-2010	19-1-2010	19-1-2011	20-1-2011

في هذا الجدول ، لاحظ أن مهمة (مقابلة الزبون) تأخرت يوم واحد. وهذا

4.7 تحليل المخاطـــر Risk Analysis

لقد أصبح تحليل المخاطر من المواضيع المهمة في تخطيط المشاريع. فهو ذو علاقة وطيدة مع نجاح المشروع أو فشله. ومع أن أدوات وطرق إعداد المشاريع قد تطورت تطورا هائلا في الوقت الحالي ، إلا أن المشاكل والمخاطر قد تكون موجودة في المشروع مما يؤثر سلبا عليه.

وهذه المشاكل أو المخاطر قد تكون ذات علاقة بجدولة المشروع أو تقدير التكاليف أو الجودة أو المتطلبات أو غير ذلك.

وبعض من هذه المشاكل قد لا يمكن تجنبها ولكن يمكن التقليل منها.

المخاطرة RISK عبارة عن مشكلة يمكن أن تسبب في خسارة أو تهديد نجاح المشروع. وبعبارة أخرى، إنها مشكلة كامنة سيكون لها تأثير سيء على المشروع.

وهذا التأثير قد يكون على:

- تكلفة المشروع
- الجدول الزمني للمشروع
- جودة المنتوج
- النجاح التقني للمشروع

تحليل المخاطرة هو عبارة عن العملية المستخدمة في تحديد ومعالجة المخاطر لمشروع ما.

لماذا يجب علينا أن نعرف مخاطر المشروع قبل تنفيذ المشروع ؟؟

1) أولا لأن المخاطر تهدد نجاح المشروع.

2) وثانيا لأن هذه المخاطر المحتملة إذا تحققت فإن تسليم المشروع سيتأخر وتكلفته ستزداد.

أثناء إنجاز تحليل المخاطر يجب أن :

1) نعد قائمة بالمخاطر
2) نحدد احتمال كل مخاطرة
3) نحدد توابع كل مخاطرة.
4) نحدد إستراتيجية الحل لكل مخاطرة.

يبين الجدول التالي نموذجا لتحليل المخاطر.

مع ملاحظة :قراءة الجدول من اليسار الى اليمين.

المخاطرة Risk	الاحتمال Probability	التأثير effect	الحل Solution
تغيير المتطلبات	80%	critical حرج	• جاد JAD • العرض المبدئي
التدريب غير كافي	70%	سهل التدبير	• التدريب • تشجيع الدراسة
غياب العاملين	60%	حرج	• المزيد من التدريب

• تحسين أوضاع العاملين			بالمشروع
• إعادة استعمال الوحدات البرمجية • استعمل أدوات مناسبة لتقدير الزمن	حرج	60%	التأخر في تسليم المنظومة
• إعادة استعمال الوحدات البرمجية • استعمل تقدير مناسب للوقت	سهل التدبير	40%	تكلفة متجاوزة
• استخدم تحليل وتصميم وبرمجة مناسبة • استخدم جميع أنواع الاختبارات • العرض التجريبي	حرج	60%	منتوج ذو جودة متدنية

4.8 أسئلة للمراجعة

1) ما هي WBS؟ وما مزاياها؟

2) اذكر بعض الارشادات التي تساعد في تقدير الوقت اللازم لتحقيق مشروع برمجي.

3) إذا كان لديك البيانات التالية لتقدير الزمن(بالأشهر) اللازم لمشروع برمجي:

- التحليل
- تصميم
- التشفير
- الاختبار

ارسم مخطط غانت للمشروع بافتراض أن كل مرحلة تبدأ بمجرد انتهاء المرحلة السابقة لها.

4) افترض أن مشروع بناء يمر بالمراحل التالية:

- التصميم المبدئي
- الخريطة
- الهيكل
- الخدمات
- تشطيبات

5

أدوات التحـــليل
Analysis

ارسم مخطط غانت للمشروع بافتراض أن جميع المراحل تتم تتابعيا أي مرحلة بعد أخرى ولكن المرحلة الأخيرة يمكن أن تبدأ مبكرا لمدة شهرواحد.

5) استخدم جدول الخط الزمني لمشروع برمجي نموذجي . افترض أن المشروع يتكون من المراحل التالية بالأسابيع wk:

- خطة المشروع 1 wk
- تجميع البيانات 1 wk
- تحليل البيانات 2 wk
- تقرير المتطلبات 1 wk
- تصميم مبدئي 1 wk
- تصميم مفصل 3 wk
- التشفير 6 wk
- الاختبار 3 wk
- توثيق 4 wk

6) ما هي المخاطرة وما تأثيرها على المشروع؟

7) هل يعتبر تحليل المخاطر ضروريا في تخطيط المشروع؟ لماذا؟

8) اذكر 4 مخاطر تواجه إعداد المنظومات؟

كيف تتعامل مع مخاطرة تغيير المتطلبات؟

5.1 مقدمــة

سنقوم في هذا الباب بعرض مجموعة من الأدوات المستخدمة في مرحلة التحليل حيث يلعب محلل النظم دورا مهما في تحديد متطلبات المستخدم واحتياجاته وذلك باستخدام أدوات وتقنيات مثل مخطط انسياب البيانات وقاموس البيانات وشجرة القرار.

في البنود التالية نصف بعض الأدوات التي يجب على محلل النظم إتقانها.

5.2 مخطط انسياب البيانات DFD

يعتبر مخطط انسياب البيانات Data Flow Diagram إحدى الأدوات المهمة في مرحلة التحليل لحصول على وصف مفصل للنظام القائم و المقترح.

فهي أداة تحليل رسومية تستخدم لتوضيح انسياب البيانات خلال النظام، وبين النظام والبيئة المحيطة به.

وهي تستعمل الرسومات لتوضيح مصدر ونهاية البيانات، وكذلك المعالجة وأماكن تخزين هذه البيانات.

ملاحظة :

لاحظ الفرق بين مخطط انسياب البيانات والمخطط الانسيابي.

ففي الوقت الذي نجد فيه أن DFD يبين انسياب البيانات والعمليات في نظام ما، نجد أن المخطط الانسيابي يبين انسياب التحكم. بعبارة أخرى، فإن DFD هو أداة تحليل بينما المخطط الانسيابي هو أداة تصميم.

خصائص مخطط انسياب البيانات :

1 يعتبر مخطط انسياب البيانات وسيلة اتصال بين المحلل والزبون أو المستخدم.

2 يمكن استعماله كأساس لمرحلة التصميم.

3 يعتبر وسيلة اتصال بين المحلل والمصمم.

4 يصف ماذا يحدث وليس كيف يحدث.

5 يستخدم لتوضيح العلاقة بين الإدخال والإخراج والمعالجة ومخازن البيانات التي تكوّن النظام.

رموز DFD :

عبارة عن أشكال هندسية مثل الدوائر والمستطيلات تعبر عن البيانات وأماكن تخزينها والمعالجة . ويوجد بالسوق العديد من البرمجيات لرسم هذه الرموز .

ونستخدم منها في هذا الكتاب برنامج VISIO لشركة مايكروسوفت لرسم مخطط انسياب البيانات . ويمكن لمحلل النظم أن يستخدمها لرسم عدة أنواع أخرى من المخططات مثل المخطط الهيكلي.

وهناك أدوات أخرى تم إعدادها من قبل شركات متخصصة، مثل:

SmartDraw

Pacestar

وهذه الأدوات يمكن تحميلها عن طريق الانترنت وهي إما مجانا لمدة معينة أو لمدة دائمة.

والآن لنستعرض هذه الرموز :

ملاحظة : هناك طرق مختلفة لرسم أشكال DFD ، وقد يستعمل المحللون رموزا مختلفة لتوضيح نفس المعالجة كما سيتضح لاحقا..

❖ سهم انسياب البيانات Data Flow

السهم يبين تدفق البيانات في مخطط DFD . وهو قد يكون خطا مستقيما



أو منحنى

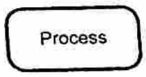


وفي كلتا الحالتين يجب أن يبين السهم اتجاه تدفق البيانات وكلمة ORDER

تعنى هنا طلبيه وتشمل أسم الصنف والكمية المطلوبة.

❖ رمز العملية (أو المعالجة)

وهو يصف عملية معالجة على البيانات . ويأخذ شكل مستطيل أركانه مقوسة كما يلي:.



أوشكل بيضاوي كما يلي:,



ويتم كتابة العملية داخل المستطيل أو الشكل البيضاوي كما سيتضح في الأمثلة اللاحقة.

❖ **مخزن البيانات Data Store**

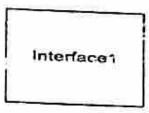
وهو مكان تخزين البيانات، ويأخذ شكل صندوق مفتوح من جهة واحدة أو جهتين كما يلي :





يمثل مخزن البيانات مكانا تكون فيه البيانات في حالة سكون. وهو قد يكون ملف حاسوب (أو قاعدة بيانات) أو ملف يدوي (مثل بطاقات الفهرس في مكتبة) .

❖ **صندوق المصدر أو النهاية Source / Destination**



صندوق المصدر أو النهاية هو عبارة عن كيان خارجي تنطلق منه البيانات (ويسمى في هذه الحالة مصدر) أو تنتهي عنده المعلومات (وتسمى في هذه الحالة نهاية أو غاية) . وكلا من المصدر والغاية يعتبران نظامين مستقلين. ويسمى هذا الصندوق أيضا بالبينية أو المؤثر .

مخطط انسياب البيانات الفيزيائي والمنطقي

يوجد نوعان من هذا المخطط هما :

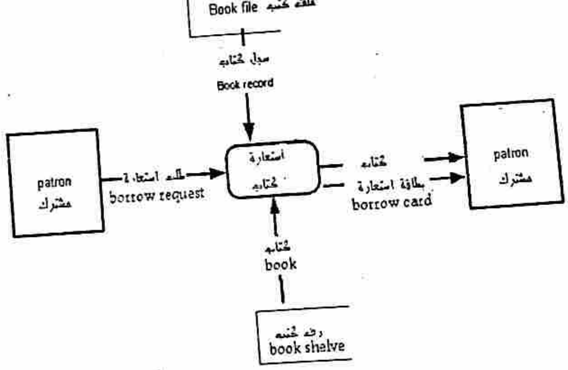
❖ مخطط انسياب بيانات فيزيائي physical DFD

❖ مخطط انسياب بيانات منطقي logical DFD

النوع الفيزيائي من DFD يبين النظام الحالي بأشيائه الفيزيائية (الملموسة) كما يراها المستخدم.

أما النوع المنطقي فيبين النظام كما يراه محلل النظم والمصمم، أي بدون عناصر فيزيائية (ملموسة) أي معلومات فقط.

مثال لمخطط انسياب بيانات فيزيائي ومنطقي لجزء من منظومة مكتبة



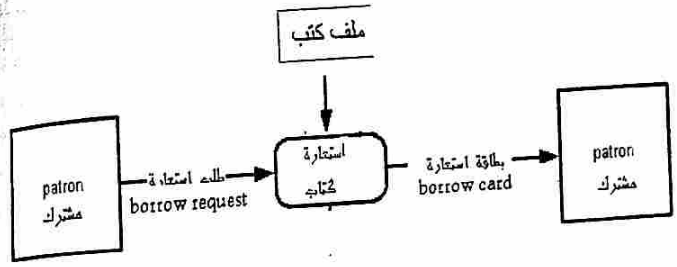
Physical DFD مخطط فيزيائي

الشكل (5.2.1) مخطط فيزيائي

في هذا الشكل هناك رف الكتب ويحتوي على كتب موجودة في المخزن وهي أشياء ملموسة ويعبر عنه بالنظام الملموس Physical System أمّا ملف الكتب وهو اما ورقي أو الكتروني في قرص صلب فيعبر عن معلومات تخص كل كتاب مثل عنوان الكتاب وأسم المؤلف وغيرها لهذا عند إعداد منظومة يسمى Information System أو مايعبر عنه بالنظام المنطقي Logical System.

ملاحظة: كقاعدة عامة ، يجب أن يتم رسم المخطط الفيزياتي قبل المنطقي. ولكن في الأمثلة التالية نتجاهل المخطط الفيزيائي لغرض التبسيط.

وعند حذف الأشياء الملموسة كرف الكتب مثلا فسنحصل على الشكل المنطقي ل DFD والذي يتعلق بالمعلومات فقط كما في الشكل التالي:



Logical DFD مخطط منطقي

الشكل (5.2.2) مخطط DFD منطقي

لاحظ في الشكل أعلاه الفرق بين المخططين.

في المخطط الفيزيائي نرى تدفق الأشياء المادية (كتب) بينما يظهر المخطط المنطقي تدفق المعلومات فقط.ولهذا السبب نهتم بإعداد نظام المعلومات Information System وهو بالطبع النظام المنطقي.

ولرسم المخطط بالكامل ، يجب عمل المخططات الآتية تباعا:

- المخطط البيئي context diagram
- مخطط فيزيائي للنظام الحالي
- مخطط منطقي للنظام الجديد

مع ملاحظة تفكيك النظام الى أجزاء من أجل التبسيط كما سيتضح لاحقا.

ملاحظة: في هذا المثال (منظومة المكتبة) تم تبسيط المخطط لغرض تسهيل الفهم.

كيف ترسم مخطط انسياب البيانات؟

يقوم محلل النظم عادة بالخطوات التالية قبل الحصول على المخطط النهائي:

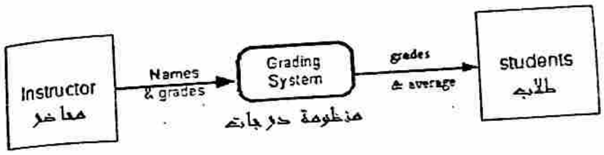
1- المخطط البيئي context diagram

يبين هذا المخطط النظام بأكمله كعملية (معالجة) واحدة متصلة بالكينونات الخارجية. وبذلك فهو يعتبر أوسع صورة للنظام. ويبين حدود النظام ونطاقه.

لاحظ أن المخطط البيئي يمكن فهمه بسهولة من قبل المستخدم.

مثال لمخطط بيئي.

ارسم مخطط بيئي لمنظومة درجات الطلبة Grading System في كلية ما حيث يستطيع الطالب عن طريقها الاطلاع على متوسط درجاته Average بعد أن يقوم الأستاذ بإدخالها.



لشكل (5.2.3) المخطط البيئي لمنظومة الدرجات

لاحظ في هذا المخطط أن

- مصدر البيانات هو المحاضر.(أو المدرس)
- مستهدف البيانات هو انطالب
- النظام نفسه يمثل العملية (المعالجة) الأساسية.

مثال

ارسم مخططا بيئيا Context Diagram لمنظومة رواتب Payroll System موظفي الجامعة حيث تشمل المدخلات:

❖ بيانات الموظفين Emp. Data من قسم شؤون الموظفين Emp Dept

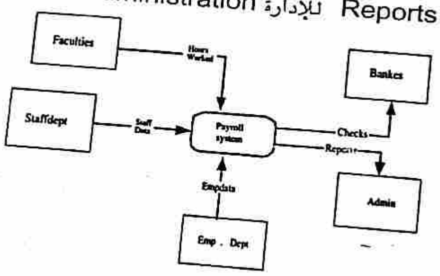
❖ بيانات هيئة التدريس Staff Data من قسم شؤون هيئة التدريس Staff Dept

❖ عدد الساعات Hours Worked الإضافية من الكليات Faculties

أما مخرجات النظام فتشمل :

❖ صكوك المرتبات Checks موجهة للمصارف Banks

❖ تقارير Reports للإدارة Administration



الشكل(5.2.4) مخطط بيئي لمنظومة رواتب الجامعة

checks	صكوك	payroll system	منظومة مرتبات
reports	تقارير	faculties	الكليات
hours worked	ساعات العمل	admin.	الإدارة
emp.data	بيانات الموظفين	banks	المصارف
Emp. dept	قسم الموظفين	staff dept.	قسم هيئة التدريس

2- مخطط المستوى 0 (Level 0)

يكون هذا المخطط عادة أكثر تفصيلا من المخطط البيئي . وتتمثل التفاصيل أساسا في إضافة مخازن البيانات.

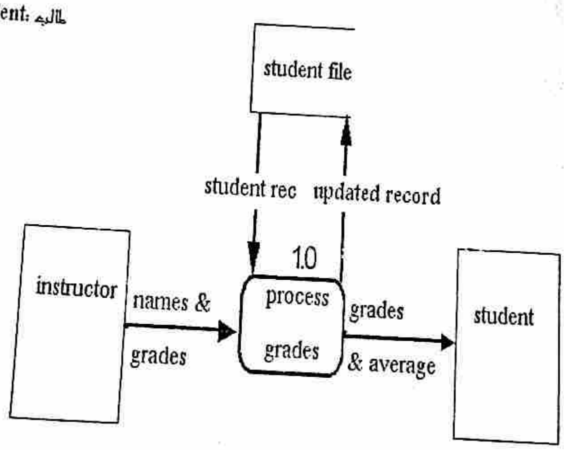
ويجب أن يكون اسم العملية (أي منظومة الرواتب) الآن أكثر تحديدا.

مـــثال

ارسم مخطط المستوى 0 للمثال أعلاه حيث يتم تخزين البيانات كلها في ملف واحد يسمى ملف الطالب.

يبين الشكل 5.2.5 مخطط المستوى 0 المطلوب. ونجد هنا أننا أضفنا (ملف الطالب) للمخطط البيئي وغيرنا اسم العملية من (منظومة رصد درجات) إلى الأمر الأكثر تحديدا : (Process Grades عالج رجات).

updated record: سجل معدل
student record: سجل طالب
process grades: معالجة درجات
student file: ملف طالب
istructor: محاضر
student: طالب



الشكل(5.2.5) مخطط المستوى 0 لمنظومة الدرجات

لاحظ أيضا أن الأسهم بين العملية والملف في اتجاهين لأن البيانات تنساب من وإلى الملف.

3- مخطط المستوى 1 (Level -1)

تتجزأ العملية الرئيسية هنا إلى عمليات مفصلة لكي يتضح النظام بصورة أكثر تفصيل . لذلك يسمى مخطط DFD أحيانا بمخطط الفقاعة حيث كل عملية (معالجة) تشبه الفقاعة التي تنقسم إلى فقاعات (عمليات أصغر)وهو مبدأ التجزئة من أجل التبسيط.

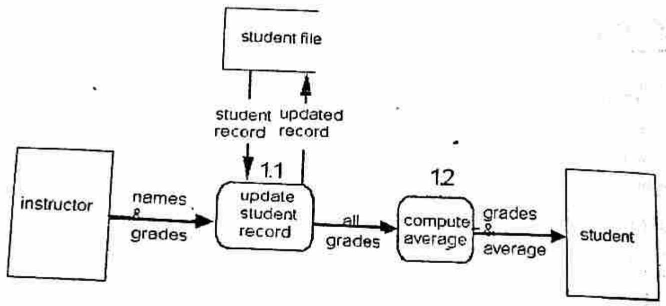
مثال

ارسم مخطط المستوى 1 للمثال (درجات الطالب) حيث تقسم العملية الرئيسية(عالج الدرجات) إلى عمليتين فرعيتين (وظيفتين فرعيتين):

1.1عدّل سجل الطالب Update Student Record

1.2 احسب المتوسط Compute Average

كما في الشكل التالي:



الشكل(5.2.6) المستوى 1 لمنظومة الدرجات

ملاحظــة :

في الشـكل 5.2.6 لـدينا عمليتـان فرعيتـان فقـط. في الواقـع يجـب أن يكون هناك 3 عمليات فرعية كما يلي:

1- استجلاب سجل طالب
2- تعديل سجل طالب
3- حساب المتوسط

حاول رسم DFD لهذا النظام باضافة عملية أستجلاب السجل لعمليات تعديل السجل وحساب المتوسط فتصبح العمليات كالتالي:

1.1 جلب سجل الطالب

1.2 تعديل سجل الطالب

1.3 حساب المتوسط

- شروط رسم مخطط انسياب البيانات :

لرسم مخطط انسياب البيانات جيد يجب إتباع النقاط والإرشادات التالية:

1. ابتعد عن التفاصيل في المخطط البيئي والمستوى 0
2. ابدأ الرسم من اليسار إلى اليمين ومن الأعلى إلى الأسفل.
3. عنون كل صندوق معالجة بفعل أمر ومفعول به مثل "احسب الفاتورة"
4. لاتبين المعدات أو الأشياء المادية في المخطط المنطقي.
5. يمكنك تكرار مخازن البيانات والكيانات الخارجية لغرض تجنب التقاطعات والحصول على وضوح الأفضل.
6. عند التجزئة قد يحتوي المستوى Level على عدد من العمليات (من 2 إلى 7) .
7. عنون انسياب البيانات على كل سهم.
8. ابدأ برسم أولي ثم اعمل مراجعة له مع زميلك في العمل والزبون للتأكد من أن المخطط يمثل فعلا النظام تحت الدراسة.
9. يمكن أن يكون للمخطط مستويات مختلفة من حيث التفاصيل (أي التجزئة).

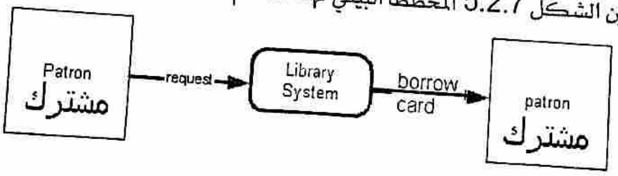
10. بعض العمليات تحتاج إلى تجزئة وعادة ما نحتاج إلى عملية ترقيم لتوضيح مستوى التجزئة. تسمى هذه الطريقة بتحديد المستويات.
11. هذا المخطط لا يبين حلقات أو عمليات حسابية أو قرارات.

مثـال : منظومـة مكتبـة

في هذه المنظومة يقوم المشترك أو المستعير patron بطلب استعارة كتاب معين مستخدما نموذج طلب كتاب Request ، وعليه تكون أول عملية معالجة (process) هي البحث عن الكتاب.مع ملاحظة ان هذا المخطط من النوع المنطقي وليس الفيزيائي لأننا لم نوضح الكتاب كشي ملموس.

ويتم البحث في ملف الكتب book-file . في حالة وجود الكتاب تتم إعارة الكتاب book-borrowing للزبون وتسجيل ذلك في بطاقة الإعارة.

يبين الشكل 5.2.7 المخطط البيئي لهذا النظام.

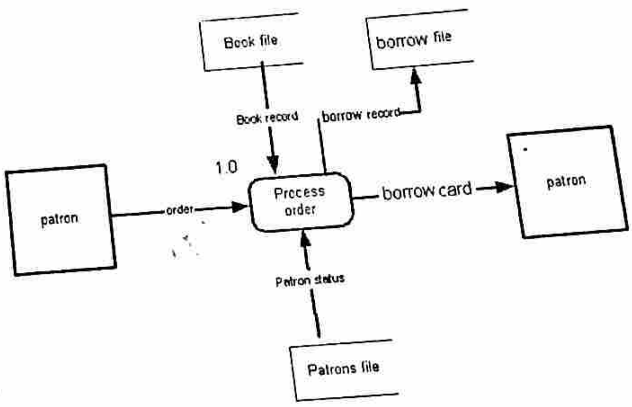


الشكل 5.2.7 المخطط البيئي لنظام مكتبة.

المستوى 0 في هذا المثال موضح بالشكل 5.2.8 حيث قمنا بتغيير العملية (منظومة مكتبة) إلى (معالجة طلبية) و أضفنا 3 مخازن بيانات (ملفات):

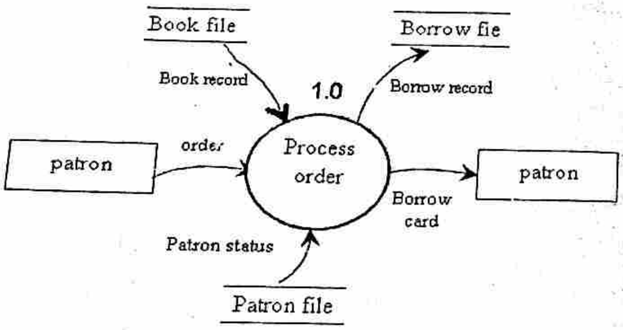
- ملف الكتب Book File
- ملف الاستعارة Borrow File
- ملف المشتركين Patron File

معنى Patron Status : حالة المشترك ويشير ماذا كان المشترك في القائمة السوداء. في هذا المخطط افترضنا وجود الكتاب المطلوب وذلك لغرض تبسيط الشكل. لاحظ أننا عنونا المعالجة بالرقم 1.0 .



الشكل(5.2.8) المستوى 0 لمخطط منظومة مكتبة

كما لاحظنا سابقا فإن الرموز قد تختلف في الشكل من DFD إلى آخر. لتأكيد هذه النقطة نعيد رسم المخطط في الشكل 5.2.8 كما في الشكل (5.2.9).



الشكل(5.2.9) المستوى 0 لمخطط منظومة مكتبة

لاحظ في هذا المخطط استخدام الشكل الدائري لعملية المعالجة بدلا من المسطيل المقوس الأركان ، وكلاهما صحيح.

المستوى 1 لمنظومة المكتبة

دعنا ألان نحاول تقسيم المستوى 0 من منظومة مكتبة (كما هو مبين بالشكل 5.2.9) الى المستوى 1 .

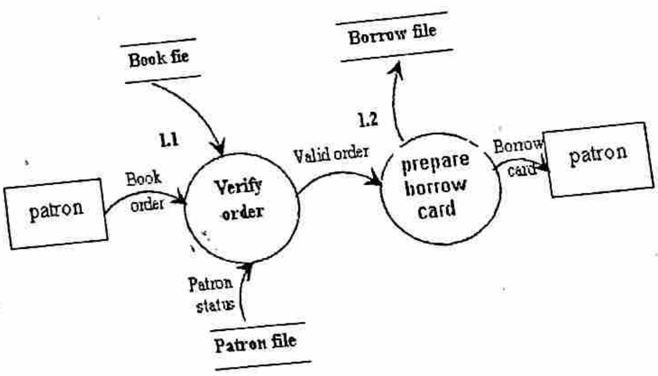
العمليـة 1.0 (عـالج الطلبيـة Process Order) تجـزأ إلى عمليتين فرعيتين (لغرض التبسيط) هما:

1.1 Verify book order التأكد من وجود الكتاب

1.2 Prepare borrow card. إعداد بطاقة الاستعارة.

ويبين الشكل 5.2.10 العلاقة بين عمليات المعالجة.

من الواضح أن هذا التقسيم يعطي فهما أوضح للنظام تحت الدراسة.



الشكل(5.2.10) المستوى 1 لنظام مكتبة

لاحظ في الشكل(5.2.10.) الحاجة إلى مزيد من التفاصيل. مثلا قد نحتاج إلى العمليات التالية لغرض تفصيل انسياب البيانات للنظام بدقة أكثر:

- .تحقق من وجود الكتاب.
- إذا لم يكن موجودا ، اعرض رسالة بذلك.
- جهّز سجل زبون في حالة عدم وجوده
- جهز سجل استعارة لتخزينه في ملف الاستعارة.
- جهز بطاقة استعارة لتسليمها الى المشترك.

نقترح أن يقوم الطالب الآن برسم DFD ذي مستوى- 2 لهذا النظام متضمنا العمليات السابقة..

مثال : منظومة الموظفين

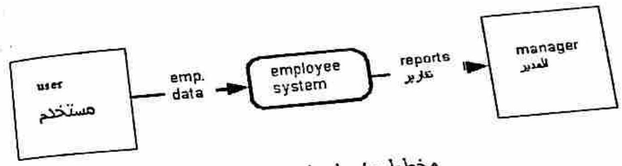
ارسم مخطط (DFD) لمنظومة الموظفين التي تقوم بالأعمال لتالية:

1) إضافة موظف جديد

2) تعديل سجل موظف

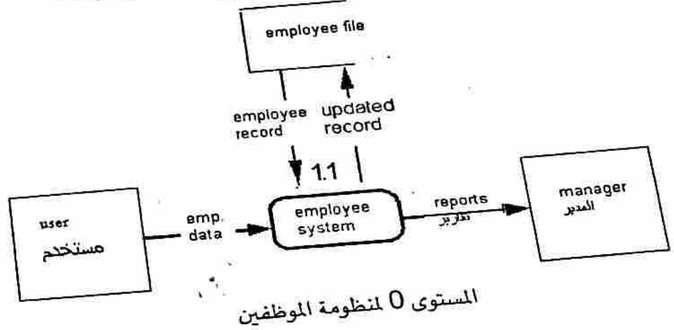
3) الاستفسار عن موظف

لاحظ أن هذه العمليات لا يلزم إنجازها تسلسليا بل يمكن أن تنجز عملية واحدة فقط في كل مرة حسب الاختيار.والشكل التالي يبين المخطط البيئي للمنظومة.



مخطط بيني لمنظومة موظفين

لاحظ في DFD المستوى -0 وجود ملف الموظفين Employee- File.

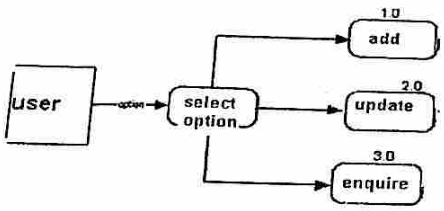


المستوى 0 لمنظومة الموظفين

الان لتجزئة المنظومة نبدأ برسم المستوى- 1 والذى يبين معاملات المنظومة:

- إضافة سجل موظف Add
- تعديل سجل موظف Update
- استفسار عن بيانات موظف Enquire

حسب الشكل التالي:



user المستخدم
select option حدد الاختيار
enquire (about an employee) استفسر عن موظف
update (employee record) عدّل سجل موظف
add (employee record) اضف سجل موظف

المستوى1 لنظام سجلات الموظفين.

ويوضح المثال المبين أعلاها مخطط انسياب البيانات لمنظومة الموظفين حيث تعتمد المعاملة على اختيار المستخدم.

بعض الكتب لا تستعمل مخطط المعاملات بل تستعمل فقط مخططات التحويل(التتابعي).

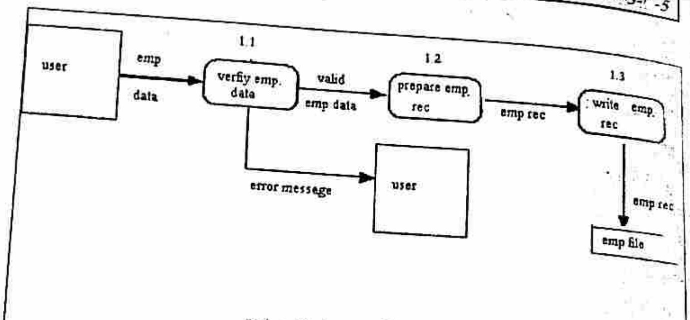
نحن نفضل استعمال في المستوى 1 هذه الطريقة لأن العديد من المنظومات من نوع المعاملات..

بعد ذلك يتم تجزئة كل عملية (معاملة) إلى المستوى 2 أو أعلى.

دعنا الآن نرسم مخطط المستوى 2 لعملية {إضافة موظف} كما هو مبين بالشكل التالي

لاحظ المصطلحات التالية المستخدمة في مخطط : DFD

- المستخدم
- بيانات موظف — Emp. Data
- تحقق من بيانات الموظف — Verify Emp. Data
- بيانات موظف صحيحة — Valid Emp.Data
- جهّز سجل موظف — Prepare Emp. Rec
- اكتب سجل موظف — Write Emp. Rec
- ملف الموظفين — Emp. File



المستوى 2 لعملية الإضافة

على الطالب أن يرسم مخطط DFD مماثلا لتجزئة عملية { تعديل سجل موظف} كما يلي:

❖ تعديل سجل موظف : Update
- أجلب السجل Retrieve Rec
- غيّر السجل Replace Rec
- أعد كتابة السجل Rewrite Rec

ولشرح خطوات كل عملية معالجة في مخطط DFD يمكننا استخدام أداة الانجليزية المركبة(الهيكلية) كما سنرى لاحقا.

5.3 الانجليزية المركبة Structured English

الإنجليزية المركبة (تسمى أيضا الانجليزية الهيكلية) هي أداة تستخدم في التحليل لوصف سياسات أو إجراءات نظام أو المنطق لمعالجة كل عملية في مخطط انسياب البيانات بلغة إنجليزية دقيقة نصيا.وبعبالرة أخرى هي مجموعة الخطوات(أي الخوارزمية)بلغة المستخدم.

> **ملاحظة :** لاحظ أن هذا المصطلح SE يمكن أن يطبق على أي لغة أخرى غير الانجليزية مثلا اللغة العربية.
> وفي هذه الحالة نسميها العربية المركبة وهي ضرورية في حالة أن المستخدم لا يتقن اللغة الانجليزية.

استعمالات الانجليزية المركبة :

تستدخدم الانجليزية المركبة في المسائل التي تدمج بين سلسلة من الأعمال والحسابات مع اختبارات وتكرارات.

قواعد كتابة الانجليزية المركبة :

1) وضح المنطق كمجموعة تعليمات خطوة خطوة بما في ذلك الحسابات وتراكيب القرارات.
2) استعمل مصطلحات من قاموس البيانات ومخطط انسياب البيانات.
3) يمكن التعبير عن منطق العمليات في النظام بمجموعة من التراكيب التتابعية والتكرارية والاختبارات.

4) استعمل التنسيق والمصطلحات الخاصة في كتابة جمل انجليزية لوصف منطق المعالجة والسياسات.
5) استخدم حروفا كبيرة في الكلمات المفتاحية.
6) استعمل أفعال مألوفة ومناسبة.

مصطلحات الانجليزية المركبة

1. القائمة التالية تحتوي على أفعال مناسبة يمكن أن تصف عمليات في الانجليزية المركبة:

- Compute احسب
- Determine حدد
- Verify حقق
- Check افحص
- Update عدل
- Prepare جهّز
- Enter أدخل
- Generate كوّن
- Read اقرأ
- Print اطبع
- Display اعرض

- Format شكّل
- Retrieve استرجع
- Compare قارن
- Edit حرّر
- Move أنقل
- Add أضف
- Sort صنف
- Merge ادمج

وتستخدم نفس الأسماء والمصطلحات المعرفة في قاموس البيانات مثل:

- Current-balance الرصيد الحالي
- Item-price سعر الصنف
- Student-name أسم الطالب
- Number-of-students عدد الطلبة
- Date-of-last-transaction تاريخ آخر معاملة

كما نستخدم أيضا الكلمات المحجوزة المستعملة في البرمجة الهيكلية مثل:

- if-then-else لو- فإن- وإلا
- case. في حالة أن

- do–while انجز- طالما
- repeat–until كرر- حتى

والمثال التالي يبين خطوات لحساب المرتب باستخدام الانجليزية المركبة(أو العربية المركبة) كما يلي:

Input hours-worked and rate. ادخل عدد ساعات الشغل وأجر الساعة الواحدة

IF hours–worked are less than or equal 40 THEN
Compute wage by multiplying hours–worked by rate
لو عدد ساعات الشغل أقل من أو يساوي 40
فاحسب الأجر بضرب عدد الساعات في أجر الساعة الواحدة.

ELSE وإلا

Compute amount1 by multiplying rate by 40
فاحسب المقدار 1 بضرب أجر الساعة الواحدة في 40

Compute amount2 by multiplying hours- worked over 40 by 1.5*rate
واحسب المقدار 2 بضرب عدد ساعات الشغل الزائدة عن 40 في 1.5 * أجر الساعة الواحدة.

Compute wage by adding amount1 to amount2.
احسب الأجر بجمع المقدار 1 مع المقدار 2.

قد يتبادر للذهن في المثال السابق أنها خوارزمية أو شفرة برنامج والحقيقة أنها خوارزمية ولكنها مبسطة وبلغة المستخدم وفي مرحلة التحايل وليس مرحلة التصميم.

ملاحظة:

لاحظ أن الإنجليزية المركبة يمكن فهمها من قبل المحلل والزبون على حد سواء ولهذا يمكن أن تكتب بأي لغة مثل العربية(حاول كتابة العربية المركبة) والتي يمكن أن نقول أن الانجليزية المركبة ماهي الآ خوارزمية(مجموعة خطوات)أسسها العالم المسلم الخوارزمي منذ مئات السنين ويعود الفضل إليه .

مثال2

المثال التالي يبين استخدام الانجليزية المركبة لسياسة حساب التخفيضات في منظومة المبيعات:

IF customer type = "old customer" THEN
لو نوع الزبون = زبون قديم فإن
IF amount of sales is more than 10000
لو مقدار المبيعات أكبر من 10000
Then discount = 10 % فإن التخفيض=
ELSE discount = 8% وإلا فإن التخفيض=
ELSE
وإلا "زبون جديد"

IF amount of sales is more than 10000
لو مقدار المبيعات أكبر من 10000
Then discount = 5 % فإن التخفيض=
ELSE discount = 3 % وإلا فالتخفيض=

5.4 شجرة القرار Decision Tree

شجرة القرار عبارة عن أداة تحليل رسومية تستخدم لتوضيح سياسة أو علاقة في نظام.

وعادة ما يتم استعمال هذا المخطط لتوضيح الأفعال Actions التي تنفذ في الحالات Condtions (الشروط) المختلفة.

فمثلا الجملة: اذا تجتهد تنجح

- الشرط Condition فهو يجتهد
- الفعل Action فهو تنجح

استخدامات شجرة القرار:

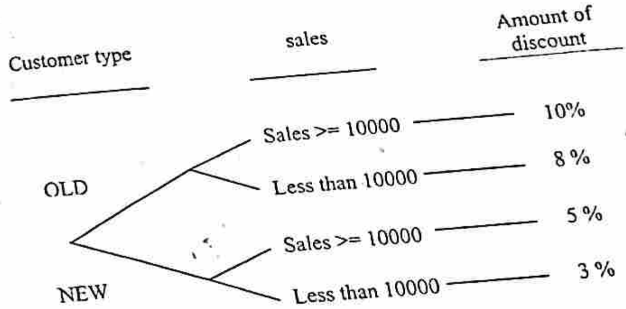
1. تتعامل شجرة القرار مع السياسات Policies ذات التفرع المتعدد مثل حساب تخفيضات المبيعات ومكافئات الانتاج وحساب الأجوروتسيب الطلبة في الجامعات.
2. تستخدم عندما يكون عدد الافعال صغيرا ويمكن توضيح كل الاحتمالات.

3. من الأفضل استخدامها للتحقق من المنطق وعندما تكون القرارات غير معقدة.

مثـــال لشجرة القرار

يخدم هذا المثال نفس الغاية لوصف سياسة التخفيض في منظومة المبيعات في مثال 2 في الفقرة 5.3 .

- نوع الزبون customer type(زبون قديم / زبون جديد)
- قيمة المبيعات sales
- قيمة التخفيضات Amount of Discount



هذا الشكل على هيئة شجرة جذرها على الشمال وتتفرع الى اليمين ويعبّر عنها في البرمجة بجملة if-Then-Else .

5.5 جدول القــــرار Decision Table

هذه الأداة يمكن أن تكون بديلة لشجرة القرار لغرض تمثيل سياسة النظام بطريقة مختصرة أي وضعها في مصفوفة ثنائية الأبعاد التي يمكن تخزينها في الحاسوب في حيز ضيق ولا تأخذ سعة كبيرة من الذاكرة ويمكن تحويلها إلى شفرة برنامج شأنها شأن شجرة القرار أو الانجليزية المركبة.

تعريف : جدول القرار عبارة عن أداة تحليل وتصميم على شكل جدول تبين المنطق الذي يربط بين الحالات والأفعال (Actions/ Conditions) وهو عادة ما يكون على هيئة مصفوفة (صفوف واعمدة).

استخدامات جدول القــــرار :

1. عندما تكون الافعال معتمدة على مجموعة كبيرة من الشروط .
2. تستخدم في تطبيقات الادارة والتحكم.
3. كما يمكن استعمالها في التطبيقات التي تتعلق بالآتي:
   - التصنيع
   - سياسة المبيعات والتخفيضات
   - المالية
   - البحث والتطوير
   - منظومة تنسيب طلبة الشهادة الثانوية إلى الجامعات.

مكونات جدول القـرار :

❖ الشروط **Conditions** : حقائق أو أحداث تقرر الأفعال التي يجب أن تنفذ.

❖ الأفعال **Actions**: العمليات أو المعالجات التي تنفذ تحت شروط معينة.

❖ قواعد القرار **Decision Rules** : تعبر عن العلاقة بين الشروط والأفعال.

الجدول التالي يمثل الشكل العام لجدول القرار:

	قواعد القرار Decision Rules					
شـــروط conditions						
أفعـــال actions						

مثال **1** : المطلوب إعداد جدول القرار لحساب مرتب موظف حيث المرتب يساوي مجموع المرتب الأساسي ومكافأة العمل الاضافي وهي تحسب فقط إذا كانت ساعات العمل hours-worked أكبر من 40 ساعة.

Conditions	Decision rules		
	1	2	3
Hours – worked < 40 Hours – worked = 40 Hours – worked > 40	Y N N	N Y N	N N Y
Actions			
Compute basic pay Compute over-time	X -	X -	X X

الجدول أعلاه يمثل أستعمال جدول القرار لتوضيح سياسة حساب راتب موظف.

لاحظ في هذا الجدول استعمال الرموز التالية:

Y : Yes

N : No

X : Applicable قابل للتطبيق

- : Not Applicable غير قابل للتطبيق

لاحظ أيضا أن عدد قواعد القرار يعتمد على كل من عدد الشروط وعدد الأفعال.

في المثال المذكور أعلاه لدينا 3 شروط و فعلين اثنين وثلاثة قواعد قرار.

والمثال التالي يبين أن عدد قواعد القرار يمكن أن يكون أكثر من عدد الحالات.

مثــال 2 :

المطلوب اعداد جدول قرار لتوضيح سياسة مبيعات حيث تعطي معاملة خاصة للزبائن الذين يتوفر فيهم شرطان من الشروط التالية:

- قيمة المشتريات تتجاوز 10000 دينار.
- الدفع في الوقت المحدد
- اكثر من 3 سنوات كزبون

في هذا المثال نلاحظ أن الحالات الثلاثة يمكن أن تتوفر كلها في زبون واحد أو بعض منها.

وفي الواقع ، عدد قواعد القرار في هذا المثال هو 2 أس 3 أي 8.

وبصورة عامة ، إذا كان لدينا n حالة مستقلة، فإن

Number of decision rules = 2^n

عدد قواعد القرار

وبذلك يكون جدول القرار كما يلي:

Conditions	Decision Rules							
Purchase > 1000	N	Y	N	Y	N	Y	N	Y
On-time payment	N	N	Y	Y	N	N	Y	Y
Year > 3	N	N	N	N	Y	Y	Y	Y
Actions								
Normal	X	X	X	-	X	-	-	-
Special	-	-	-	X	-	X	X	X

مزايـــا جداول القـــرار

1. وسيلة لوصف سياسة النظام.
2. تساعد في التصميم والتشفير.
3. يمكن استعمالها في التوثيق لتلخيص البيانات والمعلومات .
4. تخزين جدول القرار كمصفوفة في داكرة الحاسوب.

يكمن عيبها في صعوبة تكوينها خاصة لمن يستعملها لأول مرة.وأيضا قد تكون أسهل للمبرمج في فهمها من المستخدم. لهذا فهي وسيلة تواصل أكثر بين المحلل والمبرمج.

والجدول التالي يبين مقارنة بين الادوات الثلاثة من حيث سهولة الاستعمال والتركيب المنطقي.

مقارنة بين شجرة القرار وجدول القرار والانجليزية المركبة.

معايير المقارنة	Decision tree شجرة القرار	Decision Table جدول القرار	Structured English
بساطةالاستعمال	v. good جيد جدا	Poorضعيف	Moderate متوسط
Programming البرمجة	**Moderate** متوسط	Good جيد	v. good جيد جدا
Changeability قابلية التغيير	Moderate	Poor ضعيف	Good
Logic structure التركيب المنطقي	v. good	Moderate	Good
User verification تحقيق المستخدم	Good	Poor	Moderate

ستطيع أن نقول أن جداول القرار تستخدم عندما يوجد عدد كبير من
لأفعال والحالات. بينما تستخدم أشجار القرار عندما يوجد عدد بسيط من
لأفعال والحالات ويكون كل شرط محتملا.

دراسة مقارنة لدواعي استخدام أدوات التحليل لوصف سياسة عمل المنظومة:

Decision tree	Decision table	Structured English
• تستعمل للقرارات البسيطة • تستعمل للتأكيد المنطقي.	• تستعمل للمسائل ذات المنطق المركب من عدة شروط.	• تستعمل للمسائل التي تحتوي على الحلقات والقرارات والسلسلة من الأفعال.

من هذا الجدول نستخلص أنه إذا كانت المسألة التي لدينا بسيطة يمكننا أن نستعمل شجرة القرار. أما إذا كانت المسألة معقدة فنستعمل إما جدول القرار أو الانجليزية المركبة.

5.6 قاموس البيانات Data Dictionary (DD)

قاموس البيانات أداة تحليل وتصميم تستخدم لحفظ انسياب البيانات ومخازن بيانات (ملفات) في مخطط انسياب البيانات DFD.

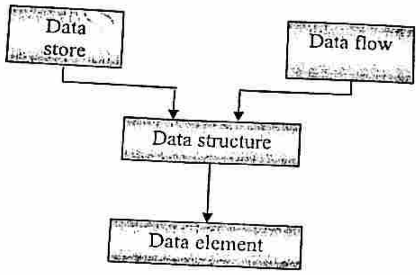
بتعبير آخر فان قاموس البيانات أداة تستخدم لتعريف وتسجيل كل عنصر بيانات في المنظومة.

الفــرق بين مخزن البيانات وانسياب البيانات في مخطط DFD

- انسياب البيانات هو تركيبة بيانات في حركة (بيانات حركية).
- مخزن البيانات هو تركيبة بيانات في سكون (بيانات ساكنة).

وفي الشكل التالي نبين أن مخزن البيانات وانسياب البيانات هما أساس تركيبة البيانات، حيث تتكون تركيبة البيانات من عناصر بيانية نبدأ من أسفل الى أعلى حيث مجموعة عناصر البيانات Data Elements تكوّن تركيبة البيانات Data Structure وأن تركيبة البيانات امّا أن تكون مخزن بيانات Data Store أو انسياب بيانات Data Flow.

هرمية وصف البيانات



Data Flow انسياب بيانات
Data store مخزن بيانات
Data structure تركيبة بيانات
Data element عنصر بيانات

ملاحظات :

1. قاموس البيانات يمكن كتابته على ورق أو الكترونيا Softcopy كجزء من نظام إدارة قاعدة البيانات (مثل أوراكل أو SQL).
2. قاموس البيانات هو أساس تصميم قاعدة البيانات التي يتم تجهيزها في مرحلة التصميم حيث تصف كل عنصر بيانات.
3. لاحظ أيضا أن الجزء الرئيسي من قاموس البيانات هو عنصر البيانات(متغيرأو حقل).

5. مصدر عنصر البيانات Source

المصدر : يمكن أن يأتي عنصر البيانات من مصادر مختلفة مثل أداة إدخال (مثلا يمكن إدخال رقم صنف بضاعة من لوحة المفاتيح) أو جداول

خصائص عنصر البيانات :

يجب أن يحتوي عنصر البيانات في قاموس البيانات على الخصائص التالية:

1. اسم عنصر البيانات Data Element name

الاسم : اسم وحيد يعين لعنصر بيانات. ويفضل أن يكون الاسم مفهوما من قبل المستخدم مثل (عمر الطالب).

أمثلة لأسماء عناصر بيانات :

- اسم الزبون
- رقم الزبون
- قيمة العلاوة الشهرية
- سعر الصنف

2. اسم الحقل Field Nane

وهو أسم الحقل باللغة الانجليزية ليتم استخدامه في شفرة البرنامج مثل **Price** للدلالة على أسم العنصر سعر الصنف.

لاحظ أنه أحيانا نستخدم بعض الاختصارات في الاسماء لغرض توفير التخزين والوقت مثل (cust_no بدلا من customer-no) لرقم الزبون و تسمى بالأسماء المختزلة أو المختصرة.

3. وصف عنصر البيانات Description

الوصف : يعطي الوصف النصي لعنصر البيانات ليبين غرض استخدام هذا العنصر.فمثلا تاريخ المقاصة في منظومة الحسابات الجارية في مصرف ما معناه تاريخ استحقاق الصك ليصبح في الرصيد الفعلي لحساب زبون المصرف .

4. نطاق عنصر البيانات ومعاني القيم Domain

النطاق : نطاق عنصر البيانات ومعنى القيم.

قد يأخذ عنصر البيانات أي قيمة خلال نطاق معين (مثلا من 0 إلى 9999999.999) أوقد يأخذ قيمة واحدة من قيم محددة مثل:

Value	Meaning
1	
2	أعزب
3	متزوج
4	
5	مطلق
	أرملة

5. مصدر عنصر البيانات Source

المصدر : يمكن أن يأتي عنصر البيانات من مصادر مختلفة مثل أداة إدخال (مثلا يمكن إدخال رقم صنف بضاعة من لوحة المفاتيح) أو جداول

(مثل جدول الراتب الأساسي) أو من عملية حساب (مثل حساب صافي الراتب).

6. أين يستخدم عنصر البيانات Where used

هذا المصطلح يستعمل للإشارة إلى المكان الذي سيستعمل فيه عنصر البيانات. مثلا قد يستعمل عنصر البيانات (سعر الصنف) في حساب الفاتورة أو تقرير المبيعات.

7. مخزن عنصر البيانات Storage

التخزين : قد يتم تخزين عنصر البيانات في ملف أو قد يكون للاستعمال المؤقت كوسيط مثل المجموع للحصول على المتوسط.

8. طول ونوع عنصر البيانات

يجب أن يحدد قاموس البيانات نوع وطول عنصر البيانات فمثلا يكون النوع إما صحيحا أو حقيقيا أو منطقيا أو تاريخا أو زمنيا أو حرفيا...الخ. أما الطول فهو حجم عنصر البيانات الذي يقاس عادة بعدد الخانات أو عدد الحروف.

مثلا قد يكون نوع حقل البينات لعنصر البيانات اسم الطالب حرفي أو رمزي وطوله 30 حرف.

9 القيمة الافتراضية لعنصر البيانات Default Value

القيمة الافتراضية لعنصر البيانات: هي القيمة المعرفة مسبقا وذلك لغرض تقليل أخطاء الإدخال وتوفير زمن الإدخال. ويتم اختيارها عادة بحيث تكون هي القيمة الأكثر احتمالا وذلك لغرض تقليل احتمال تغييرها.

على سبيل المثال القيمة الافتراضية لتاريخ فتح حساب بالمصرف من الأفضل أن تكون التاريخ الحالي (أي تاريخ الجهاز أثناء عملية إدخال البيانات) لأنه القيمة الأكثر احتمالا.لهذا عند الرغبة في ادخال تاريخ فتح حساب مصرفي ستجد أن تاريخ اليوم هو المعروض على الشاشة كتاريخ فتح الحساب ويمكن تغيره ادا اراد المشغل ذلك.

10 هل يسمح بقيمة لاشيء؟ Allow Null

اذا كان عنصر البيانات إجباريا فلا يسمح بقيمة (لاشيء) ، ولكن إذا كان اختياريا فيمكن وضع (لاشيء) كقيمة له.

مثلا إذا كان عنصر البيانات مستخدما كمفتاح للسجل فلا يسمح بوضع قيمة (لاشيء) فيه. ستجد أن رقم الزبون لايسمح بقيمة لاشي لانه ضروري ومن المفروض ادخاله أما رقم التلكس فهو غير ضروري (اختياري) ويمكن عدم ادخاله (قيمة لاشي).

لاحظ أن جميع الخصائص تستخدم لتوضيح كل عنصر بيانات لمحلل النظم كما نحتاجها في تصميم قاعدة البيانات.ولاحظ ايضا ان جزء من هذه الخصائص تستخدم في مرحلة التحليل لازالة الغموض في عناصر البيانات

مثل وصف عنصر البيانات بينما هناك خصائص اخرى تستخدم في مرحلة التصميم مثل نوع الحقل وطول الحقل .

لذلك يعتبر قاموس البيانات أداة تحليل وتصميم. فهو المستودع لمعلومات كل عنصر بيانات في المنظومة.

لاحظ أن مخطط ERD هو أيضا أداة لتمثيل أو نمذجة البيانات .لهذا هناك علاقة بين ERD وقاموس البيانات.

فاداة ERD تعطينا اسماء عناصر البيانات للمنظومة بينما قاموس البيانات يعطينا وصف وخصائص كل عنصر بيانات.

ملاحظة : عنصر البيانات يمكن أن يكون بسيطا مثل السعر أو عنصر بيانات مركب مثل العنوان فهو يتركب من عناصر بسيطة مثل أسم الشارع والمدينة .

رمـوز قاموس البيانات

عادة نستخدم الرموز التالية في قاموس البيانات في مرحلة التحليل لتحديد ماهي اسماء عناصر سجل معين :

AND وتعني +

Equivalent وتعني =

OR وتعني |

Iteration وتعني { }

Optional وتعني O

أمثلـــة لتعريفات عنصر بيانات(بسيط أو مركب) :

Name	Definition
Quantity-on-hand الكمية	The number of items in inventory عدد الاصناف في المخزن
Customer_name أسم الزبون	first + middle + lastname الاسم الاول + الاب + اللقب
العنوان Address	Street-name + city اسم الشارع + المدينة
وضع رصيد الزبون Status	good \ bad جيد / سيئ

لاحظ أن العنصر البسيط مثل سعر الصنف وأن العنصر المركب مثل العنوان

مثـــال :

- رقم الهاتف = (رقم محلي + رقم خارجي)

- الرقم المحلي = (109...103/102/101)
- الرقم الخارجي = 9+(رقم محلي + رقم المدينة + رقم الدولة)

مثـــال :

سجل الفاتورة = الاسم + { تفاصيل الصنف } n + المجموع

تفاصيل الصنف = (رقم الصنف + وصف الصنف + السعر + الكمية + المجموع الجزئي)

لاحظ ان n هنا هي عدد الاصناف أي التكرار

مثال لقاموس بيانات لنظام مكتبة :

فيما يلي جدول لقاموس بيانات لسجل من ملف رواد المكتبة أو المشتركين تستخدم عادة في تصميم قاعدة بيانات مكتبة. ونستطيع إضافة خصائص أخرى مثل القيمة الافتراضية والمدى.

والجدول التالي هو جدول مبسط لقاموس البيانات في مرحلة التصميم.

اسم عنصر البيانات	الطول	النوع	اسم الحقل
Patron number رقم المشترك	4 digits	Integer	Patron-no
Full name of patron الاسم الكامل للمشترك	30 char	String	Name
Address of patron عنوان مشترك	40 char	String	Address

1=Student or 2=Staff طالب أو مدرس	1 digit	integer	Status

مـــزايا قاموس البيانات:

1. يستطيع المستخدم أن يحصل على قائمة بالبيانات الخاصة بالمنظومة المتوفرة.
2. يتمكن المحلل والمستخدم عن طريق قاموس البيانات من التفاهم والتواصل بشأن مفردات المنظومة .
3. حل مشكلة الغموض لمعنى المصطلحات بين المصمم والمبرمج.
4. أساس لتصميم قاعدة البيانات.

5.7 أسئلة مراجعـــة

1) ما هي استعمالات DFD؟

2) إذا كانت الوظائف المطلوب معالجتها هي:

- Add a record إضافة سجل
- Update a record تعديل سجل
- Delete record الغاء سجل
- Inquire about an item الاستعلام عن شئ ما

❖ ارسم مخطط DFD عام للنظام

❖ ارسم مخطط DFD مفصل للنظام.

3) ما هي خواص مخطط انسياب البيانات؟

4) لماذا يعتبر مخطط DFD أداة مساعدة لمحلل النظم؟

5) طور منظومة استعارة الكتب في الشكل 5.2.2 لتشمل معاملة ترجيع كتاب إلى المكتبة.

6) ما هي الانجليزية المركبة ؟ وفيما تستخدم؟

7) اكتب خمسة قواعد يجب اتباعها عند استعمال الانجليزية المركبة؟

8) استعمل الانجليزية المركبة لوصف سياسة حساب الضريبة كما يلي:

- بالنسبة لدخل أقل من 1000 دينار الضريبة = 5%
- بالنسبة لدخل من 1000 إلى 2000 دينار الضريبة 7%
- أما الدخل الذي يزيد عن 2000 فعليه ضريبة 9%.

9) ارسم شجرة القرار للسياسة المتبعة في المسألة 8.

10) كوّن جدول القرار للحالات التالية:

- أكثر من 100000 دينار سنويا.

- تاريخ جيد في الدفع
- معنا لأكثر من 10 سنوات

11) اذكر 3 مزايا لقاموس البيانات.

12) ارسم DFD نظام حساب الضريبة من المعطيات التالية:

- Input الادخال :

Income الدخل

Deductions الخصميات

- Output الاخراج :

tax due الضريبة المفروضة

- Mechanism الآلية

tax file ملف الضرائب

- Control التحكم

tax policy سياسة الضريبة

tax deadlines مواعيد الضريبة

حيث الوظائف الفرعية هي

- احسب الدخل
- احسب الخصميات

▪ احسب الضريبة

13 مامعنى مصطلح قاموس البيانات؟ أذكر 4 خصائص ضرورية تستخدم في مرحلة التصميم؟ و3 خصائص تستخدم في مرحلة التحليل؟

14. أذكر مثال للخصائص الآتية لقاموس البيانات:

- المدى (النطاق)
- القيمة الافتراضية
- كول العنصر

6

العـــــرض التجريبي Prototyping

1. مقدمة
2. ماهو العرض التجريبي
3. لماذا العرض التجريبي
4. أنواع العرض التجريبي
5. المزايا
6. العيوب
7. أسئلة للمراجعة

6.1 مقدمة

من المشاكل التي نواجهها في إعداد المنظومات أن متطلبات المستخدم لا يتم تلبيتها بالكامل بصورة عامة. ويعود ذلك إلى أن هذه المتطلبات عادة ماتكون غير كاملة أو غير دقيقة.

ولحل هذه المشكلة يجب أن نعطي المستخدم فكرة مبدئية للمنظومة التي سيجري إعدادها وذلك بإعداد منظومة تجريبية مصغرة تحاكي عمل النظام الفعلي.

وهذه الفكرة تشبه بناء نموذج لسفينة قبل البدء في بنائها فعليا حتى تكون الفكرة واضحة بلا أي غموض عما سيتم إنتاجه مستقبلا.

بنفس الطريقة، في مجال إعداد المنظومات، يمكن اعداد منظومة (تسمى العرض التجريبي prototype) في مرحلة مبكرة حتى يحصل المستخدم على فكرة سريعة ولكن واضحة عن الشكل النهائي للمنظومة.

وقد ساعد في نجاح هذه الطريقة الامكانيات الجذابة لواجهة المستخدم الرسومية والوسائط المتعددة المتوفرة في اللغات المرئية الحديثة، الأمر الذي جعل من طريقة العرض التجريبي أمرا سهلا وميسرا.

وفي هذا الباب سنناقش مفهوم العرض التجريبي كمساند لتحديد واستنباط وتنقيح والتحقق من المتطلبات.

6.2 العرض التجريبي Prototyping

كما ذكرنا سابقا فإن العرض التجريبي ماهو الآ عملية إعداد وعرض منظومة تجريبية كعيّنة للزبون تمثل المنظومة المطلوبة ولكن بشكل مبدئي.

وباستخدام العرض التجريبي(المبدئي) ، يعرض معد المنظومة على الزبون منظومة ذات وظائف محددة في مرحلة مبكرة وإن كان ذلك في مستوى منخفض من الأداء والكفاءة . الفكرة هنا هي جعل الزبون يقرر شكل المنظومة النهائي في مرحلة مبكرة حيث يقوم الزبون بمشاهدة شاشات (نماذج) وتقارير وحوارات تفاعلية للمنظومة ، والأهم من ذلك بعض الوظائف المطلوبة للنظام ثم تؤخذ ملاحظاته واقتراحاته بعناية.

والميزة هنا أن الزبون يعطى فرصة لأخذ الانطباع الأول عن المنظومة ولتغيير متطلباته في مرحلة مبكرة من المشروع حيث تعتبر تكلفة التغيير قليلة نسبيا.

وتستخدم طريقة العرض التجريبي كحل لأهم مشكلة في دورة حياة إعداد المنظومة ألا وهي نقص المتطلبات أو عدم دقتها.

6.3 لماذا العرض التجريبي؟

لتحديد وتنقيح متطلبات المستخدم بوضوح أكثر.

1. لنعطي الزبون صورة واضحة عن شكل المنتوج (أي المنظومة المطلوبة) بعرض عينات منه (مثل شاشات و تقارير وحوارات تفاعلية...)
2. يعتبر العرض التجريبي عملي اكثر عندما تكون متطلبات الزبون غير واضحة .
3. لعرض أو تسويق أو بيع منتوج برمجي وبيان الفوائد التي يمكن جنيها منه.
4. لتوضيح الجدوى الفنية للمنظومة.

6.4 أنواع العرض التجريبي

1. العرض التجريبي الساكن Static Prototyping

في هذا النوع من العرض التجريبي يتم عرض نماذج من الشاشات و التقارير على الزبون للاطلاع وإبداء الملاحظات سواء بالإضافة أو التعديل أو الحذف ولكن دون أن يتمكن من إدخال البيانات أو الحصول على تقارير مباشرة من جهاز الحاسب.

وبعبارة أخرى فإن نماذج العرض التجريبي الساكن هي ببساطة تصاميم وليست منظومات قابلة للتنفيذ وبعد دراسة هذه الشاشات والتقارير ، يستطيع الزبون أن يطلب أي تغييرات لهذه الشاشات وتصاميم التقارير وقد يكون تصميم الشاشات على ورق باستخدام قلم رصاص أو استخدام لغة مرئية مثل بيسك المرئي لتصميم الشاشات والتقارير بسيطا ولا يكلف شيئا.

2. العـــرض التجريبي المحذوف

Throw-away Prototyping

إن الغرض الأساسي من العرض التجريبي هو تحديد وتنقيح متطلبات
المستخدم والذي يتم في بداية إعداد المنظومة ذات الوظائف المحددة في وقت
قصير وباستخدام أحد لغات البرمجة(فيجوال بيسك مثلا) وبدون اهتمام
بجودة المنظومة ثم يتم تنقيحها مع الزبون ولكن يتم التخلص من هذه
العينة البرمجية بعد وضوح متطلبات المستخدم، ثم يتم الانتقال إلى المرحلة
التالية (مرحلة التصميم) في دورة حياة إعداد المنظومة.

3. نمـــوذج العرض التجريبي التطوري

Evolutionary Prototyping

الغرض من هذا النموذج هو الوصول إلى المنظومة النهائية ابتداء من
نسخة أولية بوظائف محدودة ثم إعداد عدة اصدارات متعاقبة بوظائف أكثر
حتى بتم تلبية كل المتطلبات ويتم إعداد المنظومة النهائية. وذلك يعني أن هذا
النموذج يمكن استخدامه كبديل لدورة حياة اعداد المنظومة البرمجية
.ويعنى أيضا عدم وجود توثيق كامل للمنظومة.

ويمكن تلخيص نشاطات نموذج العرض التجريبي في الخطوات التالية:

1. يتم التعرف على المتطلبات من قبل معد المنظومة بمساعدة المستخدم و الزبون..

2. يتم تصميم وتنفيذ منظومة العرض التجريبي للزبون لتقوم بأداء الوظائف المهمة (وظائف محددة). ويمكن هنا استعمال لغة برمجة مرئية مثل فيجوال بيسك لإعداد هذه المنظومة بشكل سريع.
3. يتم تجربة طريقة عمل منظومة العرض التجريبي من قبل الزبون
4. يعطى الزبون (المستخدم) فرصة استعمال منظومة العرض التجريبي ونتوقع منه في المقابل أن يعطي ملاحظات او يقترح تغييرات.
5. من التغييرات المطلوبة في الخطوة 4 يمكن لمعد المنظومة البرمجية أن يجهز النسخة اللاحقة.
6. يتم تكرار الخطوات 3 و 4 و 5 حتى يتم تحقيق متطلبات المستخدم تماما ونحصل على المنظومة النهائية.

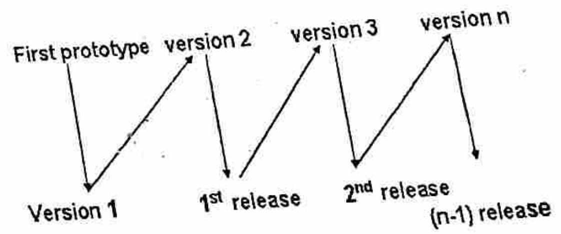
يقول أحد الرواد في هندسة البرمجيات:

(من المستحيل الحصول على منظومة تعمل من أول مرة، ويجب أن نخطط لحذف أول نسخة منها)

ويقصد بذلك أن المنظومة الأولى التي يتم عرضها على الزبون هي تجريبية ومن المتوقع التخلص منها واستبدالها بنسخة أخرى تلبي متطلبات الزبون بشكل أفضل.

لاحظ الفرق بين مصطلح "نسخة" ومصطلح "اصدار" حيث الاصدار هومنظومة تعطى للزبون للاستعمال، والنسخة هي منظومة يتم اعدادها في موقع معد المنظومة وهي مرشحة للتعديل والاستبدال بنسخة لاحقة.

لذلك يجب أن نعتبر العرض التجريبي كنسخة 1 ، والنسخة 2 كأول اصدار وهكذا..



نسخ و اصدارات المنظومة البرمجية

6.5 مزايـــا العرض التجريبي

1. يقلل من مشاكل تحديد المتطلبات.
2. يوفر مشاركة ودعم أكبر من طرف الزبون في المراحل المبكرة من المشروع (التحليل \ التصميم).
3. يقلل من المخاطرة في تأخير تسليم المشروع.
4. يمكن استعماله كقاعدة لكتابة المواصفات بجودة أعلى.
5. يمكن استعماله لتدريب المستخدمين قبل تسليم المنظومة.
6. يقلل من وقت إعداد المنظومات.
7. يقلل من التكلفة الإجمالية لإعداد المنظومات.
8. يقلل من احتمال فشل المشروع.
9. طريقة أفضل للتواصل بين المستخدم ومعد المنظومة.

6.6 عيوب العرض التجريبي

1. أحيانا يحدث التباس للزبون بأن العرض التجريبي هو المنتوج النهائي. وفي هذه الحالة يكون من الصعب إقناع الزبون أن منظومة العرض التجريبي هي للعرض فقط (منظومة تجريبية) ، وقد يطلب بالإسراع في استلام المنظومة بعد أجراء تعديلات بسيطة عليها.
2. للحصول على منظومة العرض التجريبي بسرعة ، يستخدم معد المنظومة عادة أدوات وخوارزميات بسيطة وقليلة الكفاءة وهذه

الأدوات والخوارزميات عادة ما تصبح هي الأدوات الأساسية في إعداد المنتوج النهائي مما يؤدي إلى إعداد منظومة قليلة الجودة.

3. تخلق إحساسا خادعا من التفاؤل.
4. لاينتج عنها توثيق وهذا يجعل من صيانة المنظومة أمرا صعبا .

6.7 أسئلة مراجعة

1. ما هي الفكرة الأساسية وراء منظومة العرض التجريبي؟
2. لماذا نحتاج إلى العرض التحريبي عند إعداد برمجيات جديدة؟
3. قارن بين المصطلحين " نسخة" و "اصدار" في نموذج العرض التجريبي؟
4. قارن بين أنواع العروض التجريبية.
5. أي من منظومات العرض التجريبي يستخدم أساسا في تحديد المتطلبات؟
6. هل يمكن أن يكون النموذج التطويري بديلا للنموذج التقليدي في إعداد المنظومات؟
7. صف أهم نشاطات تنجز في نموذج العرض المبدئي؟
8. اذكر 3 مساوئ للعرض التجريبي.

7

chapter seven

التحليل باستعمال مخطط حالة الاستخدام
Use Case Diagram

7.1 مقدمة

في الأبواب السابقة، كنا نتعامل بالطريقة الهيكلية (الهرمية) في التحليل، حيث استخدمنا أدوات هيكلية مثل **DFD** والإنجليزية الهيكلية. إلاّ أن الاتجاه الجديد في التحليل والتصميم يسير نحو الأسلوب الكائني {الشيئي} حيث يتم تعريف النظام على أنه مجموعة من الكائنات Objects أو الأشياء التي تكوّن النظام بدلا من مجموعة من الأجزاء البرمجية باستخدام الأسلوب الهيكلي.

وفكرة الأسلوب الكائني هي أن النظام يتكون من مجموعة من الأشياء المادية منها والمعنوية مثل التي نتعامل معها في حياتنا اليومية، وكل شي (أو كائن) له خصائص (بيانات) تصفه وعمليات (طرق) يؤديها أو تجرى عليه. أما كيفية استخدام هذا الأسلوب في البرمجة فهي إلحاق الأجزاء البرمجية (مثل الدالة أو الإجراء) القابلة للاستعمال والمخزنة مسبقا في المكتبة من مشاريع سابقة بالمنظومة الجاري إعدادها وذلك لغرض جعل إعداد المنظومة أسرع وأكثر اعتمادية ومرونة وكفاءة وقابلية للصيانة.

هذه الفكرة تشبه فكرة الدوال الجاهزة (المكتبية) المستخدمة في لغات البرمجة (مثل SQRT, SIN, EXP...الخ) التي تلحق بالبرامج عند الحاجة إليها. . بتعبير أخر، إذا افترضنا كتابة برنامج فرعى (إجراء) في مشروع إعداد منظومة سابقة فيمكن تخزين هذا الإجراء أو البرنامج الفرعي في مكتبة ثم يمكن استدعاء هذا الإجراء

أو الدالة في تطبيق مشروع مستقبلي آخر مشابه وهذا بدوره يوفر الوقت في الإعداد ويقلص الاختبار .

يمكننا أن نتتبع نشأة التحليل والتصميم الكائني كما يلي:

1978 : تعريف لغة Simula كلغة شيئية

1985 : ظهور التصميم الشيئي

1990: ظهور التحليل الشيئي

1995 : قبول ADA 95 كلغة شيئية عالية .

1996: إصدار النسخة الأولى من **UML** (لغة النمذجة الموحدة) وهي رزمة من الأدوات لإعداد البرمجيات الكائنية والتي تتكون من مجموعة من الأدوات الشيئية لاستخدامها فى التحليل والتصميم والبرمجة الشيئية.

ويوجد الآن العديد من أدوات التحليل والتصميم الشيئي.

و في هذا الباب سوف ندرس بعضا من هذه الأدوات وبالتحديد مخطط حالة الاستخدام .

7.2 الأسلوب الهيكلي والشيئي

في الأسلوب الهيكلي العلاقة بين البيانات والعمليات(المعالجة) تعتبر هشة أي مترابطة بشكل ضعيف، ويكون التركيز في الأسلوب الهيكلي فقط على العمليات (أو المعالجة) واعتبار البيانات ذات أهمية ثانوية. ولكن في الأسلوب الشيئي تدمج العمليات والبيانات(الخصائص)

في وحدة واحدة لتكوّن الكائن، وبالتالي فإن النظام يتحلل إلى مجموعة كائنات(معالجة وبيانات معا) وليس أجزاء برمجية(غالبا معالجة) كما في الأسلوب الهيكلي. وتتفاعل الكائنات فيما بينها عن طريق الرسائل.

و بتعبير آخر لا يتم الفصل بين البرامج(المعالجة) والبيانات في الأسلوب الكائني بل يتم التعامل معهما كوحدة واحدة.

7.3 ما هي التقنية الشيئية؟ Object-Oriented Technology(OOT)

تتكون التقنية الشيئية من:

- التحليل الشيئي OO Analysis
- التصميم الشيئي OO Design
- البرمجة الشيئية OO Programming
- قاعدة البيانات الشيئية OO Data Base

فالنظام الشيئي يتكون من مجموعة كائنات تسمى أشياء Objects وهذه الكائنات يتم تعريفها في مرحلة التحليل فمثلا في منظومة الخطوط الجوية تعتبر التذكرة والمسافر والرحلة كائنات أو أشياء ، ولكل كائن له خصائص Attributes تصف حالته وله إجراءات

(طـرق) Methods تصف سلوكه.فمثلا الكائن تذكرة لها خصائص مثل رقم التذكرة ونوع التذكرة وسعر التذكرة وطرق (عمليات) مثل شراء التذكرة وترجيع التذكرة.

ملاحظة :

في مصطلح النظم الشيئية : الخصائص هي البيانات (المتغيرات) أما الطرق فهي الاجراءات أو الأجزاء البرمجية .فسعر التذكرة هو خاصية وترجيع او طباعة التذكرة فهو عملية أو طريقة.

ولفهم التقنية الشيئية نحتاج معرفة المفاهيم التالية:

- الكائنات Objects
- الفصائل Classes
- الوراثة Inheritance
- الرسائل Messages
- التغليف Encapsulation
- العمليات (الطرق) Methods
- تعدد الأشكال Polymorphism

وعند استيعاب هذه المفاهيم تستطيع التعامل مع التقنية ببساطة.

الكائنات : Objects

تعتبر الكائنات في حياتنا اليومية أشياء موجودة في عالمنا الواقعي المعاش.

فالكائنات:

- زبون
- فاتورة
- طالب
- حساب مصرفي

ماهي الآ كائنات مادية مثل زبون وطالب ومعنوية مثل حساب مصرفي وفاتورة وكل هذه الكائنات بالطبع نتعامل معها في حياتنا اليومية لهذا يمكن القول أن التقنية الشيئية تقنية مألوفة وليس من صنع الخيال .

ولوصف كائن عادة ما نقوم بتحديد خصائصه وما يقوم به هذا الكائن من عمليات. أي أن الكائنات لها خصائص (خواص) تصف حالتها.

وطرق تعبر عن العمليات المطلوبة منها.

فمثلا *الزبون له* الخصائص التالية:

- رقم البطاقة الشخصية
- الاسم
- العمر
- العنوان
- رقم الهاتف

وللزبون العمليات (الطرق) التالية :

- يشتري بضاعة
- يغير بضاعة
- يستفسر عن سعر بضاعة

وكذلك الكائن الحساب المصرفي له خصائص مثل :

- رقم الحساب
- نوع الحساب
- تاريخ فتح الحساب
- رصيد الحساب

وعمليات أو طرق مثل:

- فتح حساب
- قفل حساب.
- استفسار عن رصيد حساب
- طباعة كشف الحساب

وهذا ينطبق على الكائن تذكرة في منظومة الخطوط الجوية والكائن فاتورة في منظومة المبيعات.

الفصائل : Classes

الكائنات المتشابهة يمكن تجميعها في فصيلة **class** واحدة. لذلك فإن الفصيلة تتكون من خصائص ووظائف (عمليات) يشترك فيها أكثر من كائن **object** واحد.

ملاحظة

في التحليل الشيئي نستخدم مصطلح(طريقة) Methods بنفس المعنى:

- إجراء
- وظيفة
- معاملة
- عملية

كما نستعمل مصطلح خاصية Attribute بنفس المعنى :

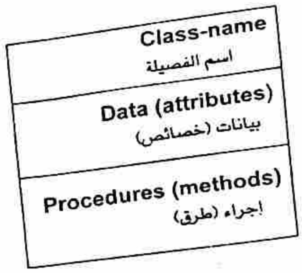
- عنصر بيانات
- حقل
- متغير

وكمثال للفرق بين الكائن (الشيء) والفصيلة نقول : أنت كائن Object وأنت تتبع فصيلة (Class) الإنسان.

والمثال التالي يبين الشكل الرسومي للفصيلة والتي تتكون من العناصر التالية:

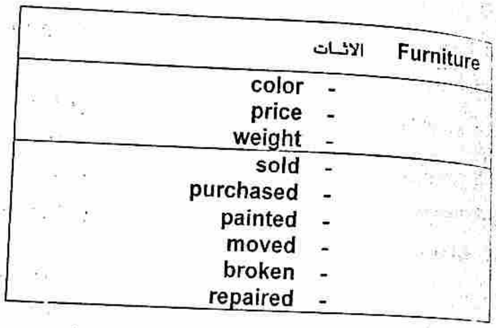
- اسم الفصيلة.
- الخصائص أو بعبارة أخرى البيانات المراد معالجتها.

- الطرق أو الإجراء وهو بعبارة أخرى الخوارزمية أو الاجراء أو الشفرة.



الشكل 7.3.1 يبين الفصيلة بالرسم

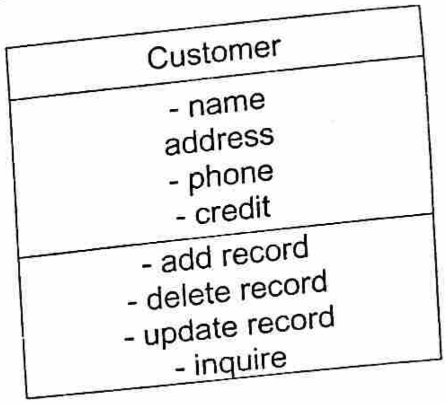
فمثلا إذا كان اسم الفصيلة (الأثاث) فربما تشمل ما يلي:
الخصائص مثل اللون والسعر والوزن والعمليات أو طرق مثل البيع والشراء والطلاء والتحريك (من المكان) والكسر والصيانة.
ويمكن تمثيل الفصيلة بالرسم كالآتي:
حيث الجزء الأول يمثل اسم الفصيلة والجزء الثاني يمثل خصائص الفصيلة والجزء الثالث يمثل عمليات او طرق الفصيلة.



الشكل 7.3.2 مثال يبين فصيلة الأثاث بالرسم

فالكرسي والطاولة هو كائنان **Objects** ينتمون إلى فصيلة **Class** الأثاث.

وكمثال آخر فإن فصيلة الزبون في منظومة المبيعات يمكن أن تكون لها الخصائص الاسم والعنوان ورقم الهاتف والرصيد الدائن ولها طرق مثل إضافة سجل زبون حذف سجل زبون وتعديل سجل زبون واستفسار عن بيانات زبون على النحو التالي:



وكمثال على خصائص كائن Object Ali Ahmed ينتمي إلى الفصيلة زبون:

Name : Ali Ahmed الاسم
Address : 34 Dahra Street العنوان
Phone : 666 6666 رقم الهاتف
Credit : 3456.75 رصيد الدين

ملاحظة:

في الأمثلة السابقة نلاحظ أن *الكائن ما هو إلا حدث فعلي ينتمي لفصيلة معينة.* ويمكن تطبيق كل الطرق التي تنتمي إلى فصيلة الزبون على هذا الكائن.

الوراثة: Inheritance

غالبا ما يتم تقسيم الفصيلةClass إلى فصائل جزئية Subclass فمثلا فصيلة السيارات يمكن أن تحتوي على الفصيلة الفرعية للسيارات الرياضية ، والفصيلة الفرعية للسيارات الاقتصادية ، والفصيلة الفرعية للسيارات الفارهة ، ...الخ.

فالفصائل الفرعية تملك كل خصائص وطرق الفصيلة الأم.

وبتعبير آخر فإن الفصيلة الفرعية ترث بيانات وعمليات (طرق) الفصيلة الأم (تسمى الفصيلة الفوقية).

ومن مزايا الوراثة أنها تسمح بإعادة استعمال سلوك الفصيلة.

فمثلا فصيلة الزبون هي فصيلة جزئية من فصيلة الإنسان.

لذلك فإن جميع خصائص وطرق فصيلة الإنسان يمكن توريثها لفصيلة الزبون.

التغليف: Encapsulation

التغليف هو شكل من أشكال إخفاء المعلومات، أي وسيلة يتم بواسطتها ربط البيانات مع العمليات (الطرق) المرافقة لها .

ويتم الوصول إلى بيانات الكائن عن طريق الرسائل Messages (البارامترات) بحيث يتم إخفاء البيانات الداخلية عن طالب البيانات ويحصل فقط على البيانات التي تهمه بواسطة الرسالة.

تعدد الأشكال :Polymorphism بولي مورفزم

مصطلح "بولي مورفزم" كلمة أصلها اغريقي تعني تعدد الأشكال فبولي تعني التعدد ومورفزم تعني الأشكال.

وفي بعض لغات البرمجة (مثل جافا) تعني وجود إجراءين لهما نفس الاسم ولكن يتم التمييز بينهما بواسطة فحص أنواع البارامترات.

فمثلا عندما نسأل عن نتيجة A+B ، فإن الإجابة تعتمد على ما هي A و B ؟ ما نوعهما؟ هل هما رقمان؟ هل هما نضيدان؟ إذا كانا رقمين صحيحين فإن الرمز + تعني الإضافة . ولكن إذا كانا نضيدين فإنه تعني الربط بين نضيدين.

وكذلك الإجراء "أحسب المساحة" فقد تعني حساب مساحة المربع أو مساحة المستطيل أو الدائرة اعتمادا على نوع البارامترات (المتغيرات)المرسلة عبر الرسالة.

7.4 مزايا التقنية الشينية

للتنقية الشيئية مجموعة من المزايا نذكر منها :

1- سهولة إعداد التطبيقات باستخدام التقنية الشيئية ، لأن الكائنات في البرمجة تمثل كائنات في الواقع الحقيقي.

2- زيادة الإنتاجية لان الفصائل مصممة بحيث يمكن استعمالها عدة مرات في منظومات مختلفة Reuse.

3- تقليص الوقت اللازم للصيانة حيث نحتاج إلى أقل تشفير في البرمجة وبالتالي تقل الأخطاء البرمجية.

5- تحسين الجودة في المنظومات التي يتم بناؤها من مكونات تم اختبارها مسبقا.

7.5 التحليل والتصميم الشيني

هناك فرق بين التحليل بصفة عامة والتحليل الشيئي، ولتوضيح ذلك نوضح الفرق بين المصطلحين حسب الأتي:

التحليل :

التحليل هو عملية تزيل الغموض عن نظام قائم، وتصف المشكلة المطلوب حلها وتحدد المتطلبات التي يجب تنفيذها للمنظومة المقترحة ، أي ما يجب على النظام عمله.

التحليل الشيني :

ينتج عن التحليل الشيئي نموذج يتكون من مجموعة من الكائنات حيث كل كائن في النموذج يمثل كائنا في العالم الحقيقي.

التصميم :

التصميم هو عملية وصف يمثل الحل أي التركيز على كيفية إعداد المنظومة ؟

التصميم الشيئي:

في التصميم الشيئي يتم تصميم المنظومة البرمجية بناء على الكائنات. ويشمل تحويل المتطلبات إلى هياكل البيانات والإجراءات الخاصة بالكائنات.

ونبدأ ألان بأحد أدوات التحليل الشيئي ألا وهو مخطط حالة الاستخدام:

7.6 مخطط وسيناريو حالة الاستخدام

- **مخطط حالة الاستخدام:** Use Case Diagram

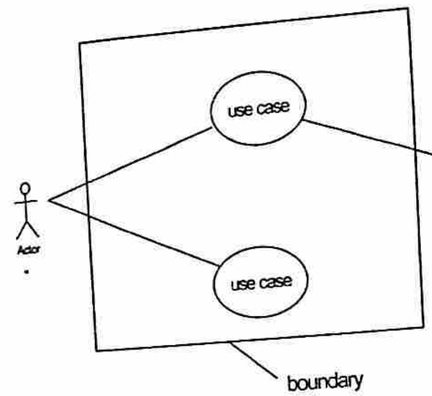
أداة تحليل شيئية رسومية تستخدم لغرض تحديد المتطلبات الوظيفي للمنظومة وهي تمثل نطاق المنظومة.

ويمكن استخدامها تقريبا في كل مشروع برمجي خاصة في التحليل كوسيلة اتصال بين المستخدم ومحلل النظم عند تعريف وتحديد المتطلبات الوظيفية.

ملاحظة: مخطط حالة الاستخدام **UCD** يمكن استخدامه في النظم الهيكلية والشيئية على حد السواء.وهي تعتبر طريقة لتحديد المتطلبات وأداة لتحليل المتطلبات.ويعتبرها المؤلف من أفضل الأدوات في مرحلة التحليل نظرا لبساطتها وفائدتها في تصميم المنظومة بالإضافة أنها يمكن أن تكون أداة تواصل بين محلل النظم والمستخدم لتحديد وظائف المنظومة المراد إعدادها. وننصح باستخدامها في تحليل أي منظومة برمجية.

ويتكون مخطط حالة الاستخدام **UCD** من حالات استخدام (وتسمى وظائف) وممثلين (مستخدمين أو نظم خارجية) في تفاعل مع بعضهم البعض لانجاز معاملات النظام.

الشكل العام لمخطط حالة الاستخدام كما يلي:



مــلاحظة :

مخطط حالة الاستخدام هـو إحدى أدوات UML في التحليل. وتعني كلمة UML لغة النمذجة الموحدة. وهـي عبـارة عـن مخططـات ونصـوص لغـرض إعـداد البرمجيات الشيئية.

رموز مخطط حالة الاستخدام :

(1) الممثل :Actor

الممثـل(الفاعـل) هــو دور يلعبــه المســتخدم أو نظــام خــارجي مــع المنظومة المقترحة.أي هو الشخص الذي يؤدي العملية أو المعاملة.

مثال :

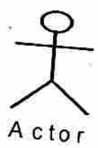
الممثل يمكن أن يلعب أحد الأدوار التالية:

- بائع
- مكتبي
- زبون
- نظام الدفع

لاحظ أن :

- الممثل الواحد يمكن ان يمثل عدة مستخدمين او انظمة خارجية فمثلا المدير يمكن أن يلعب دور الموظف في أداء المعاملات.
- المستخدم الواحد او النظـام الخـارجي الواحـد يمكـن ان يلعـب ادوار متعـددة .فمـثلا الزبـون يمكـن أن يشـتري أو يستفسـر أو يرجع بضاعة.

الرمز المستخدم للممثل أو الفاعل **Actor** هو:

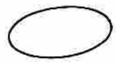


وهو رسم كاريكاتيري(كارتوني) يعبر عن إنسان يؤدي وظيفة أو معاملة.

(2) حالة استخدام : Use case

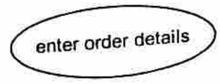
عبارة عن إجراء عن طريقه يتفاعل الممثلون مع النظام. وهو- بلغة تحليل النظم الهيكلية - عبارة عن معاملة أو وظيفة.فمثلا فتح حساب مصرفي وسحب صك وإيداع نقود تعتبر حالات استخدام أو معاملات في المنظومة المصرفية.

والشكل الذي يمثل حالة استخدام يكون بيضاويا على النحو التالي:



والإجراء(معاملة) الذي يتم تنفيذه في هذه الحالة يكتب داخل الشكل البيضوي كما يلي:



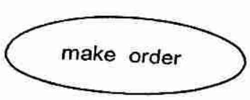


حيث :

Check Item تعني افحص الصنف

Enter Order details تعني ادخل تفاصيل الطلبية

مثال :عمل طلبية (Make Order) ترسم بهذا الشكل:



وفي حالة الاستخدام (عمل طلبية) يمكن أن تجزأ للخطوات التالية :

1- ادخل بيانات الطلبية
2- جهز الفاتورة
3- ادفع الفاتورة

لاحظ أن هذه الخطوات مرتبة تسلسليا ومنطقيا .ويعني ان حالة الاستخدام يمكن ان تجزأ الى حالات استخدام فرعية كما سنرى لاحقا.

مثال: *حالة استخدام سحب صك في نظام مصرفي*

إن حالة الاستخدام هي ملخص للخطوات التالية:

1- سلم الصك للموظف
2- التأكد من التوقيع
3- التأكد من الرصيد
4- تعديل الرصيد
5- تسليم النقد للزبون

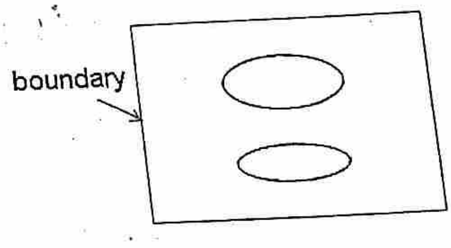
وهذه الخطوات تسمى بسيناريو حالة الاستخدام ، أي أن حالات الاستخدام في المنظومة لهم مخطط يسمى مخطط حالة الاستخدام ولكل منهم سيناريو خاص به كما سيتم شرحه لاحقا.

مـــلاحظات :

1. لكل ممثل يوجد على الأقل حالة استخدام واحدة.
2. من المهم أن نفرق بين المستخدم والممثل: المستخدم هو انسان يقوم باستخدام النظام ، أما الممثل فهواشمل ، فيمكن أن يكون الممثل مستخدما أو نظاما خارجيا يتصل بالمنظومة.

(3) الحـــدود : Boundary

الحـدود تبين نطـاق **Scope** النظـام. ويمكـن تمثيلها بخط (مستطيل) يحيط بالنظام ويفصله عن بيئته الخارجية.



لهذاوجب استخدام الشكل المستطيل ليحيط بالمنظومة او النظام ويعتبر حدودا لها ، ويعتبر كل مابداخلها ضمن نطاق المنظومة.

العلاقـــات : Relationships

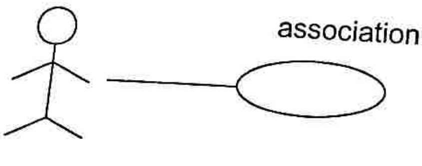
يوجد **3** أنواع من العلاقات:

1. علاقة ممثل وحالة استخدام (تسمى علاقة ربط)
2. علاقة حالة استخدام وأخرى (تسمى علاقة امتداد أو شمول)
3. علاقة ممثل بآخر أخرى (تسمى علاقة تعميم)

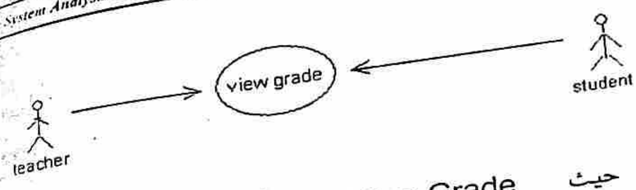
علاقـــة الربط : Association

يتم تمثيل العلاقة بين ممثل وحالة استخدام بخط مستقيم يربط بين الممثل وحالة أو حالات الاستخدام وذلك لتوضيح أن الممثل هو الذي يقوم بحالة الاستخدام أي أداء المعاملة أو الوظيفة.

والشكل التالي يبين هذه العلاقة .



والشكل التالي يبين مثال لحالة الاستخدام يقوم بها أكثر من ممثل:

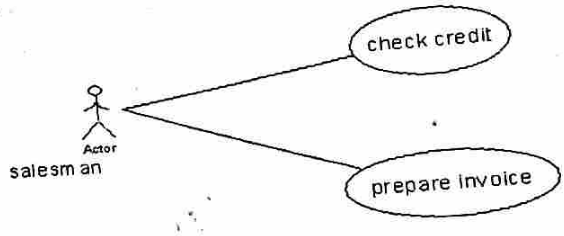


حيث View Grade معناها أعرض الدرجة

وفي الشكل التالي أيضا الممثل {البائع Salesman } يقوم بحالات استخدام متعددة:

Check credit افحص الرصيد

Prepare invoice جهز الفاتورة



- **علاقة اشتمال / امتداد :**

يمكن أن توجد علاقة بين حالة استخدام وأخرى.

وهناك نوعان من العلاقة بين حالات الاستخدام وهما

- **علاقة الشمول Include :**

علاقة الشمول تستخدم لتبين أن حالة استخدام معينة{ تسمى إما حالة الاستخدام الأساسية أو حالة استخدام الوالد} تشتمل على وظائف موجودة في حالات استخدام أخرى (تسمى حالات استخدام الأبناء).

يكون لدينا أحد الأوضاع التالية:

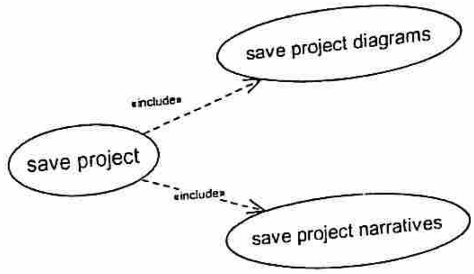
1. حالة استخدام رئيسية لديها حالات استخدام فرعية
2. حالة استخدام إضافية(ملزمة أو ضرورية) يمكن إضافتها لحالة الاستخدام الرئيسية.

ملاحظة :
يجب ملاحظة أن في بعض الأحيان نجد كلمة **use** مكافئة لكلمة **include**.
وفي هذا الفصل سوف نستخدم الاختيار**include**.

مثال 1 :

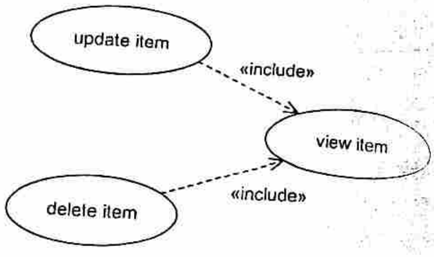
في الشكل التالي، نجد أن حالة الاستخدام (احفظ المشروع) تعتبر أساسية (حالة استخدام أب) بينما تعتبر حالة الاستخدام (احفظ خرائط المشروع) وحالة الاستخدام (احفظ نصوص المشروع) أبناء لحالة الاستخدام الأساسية(احفظ المشروع) لان المشروع يحتوي على نوعين من الوثائق : خرائط Diagrams ووصف نصي Narratives.

حيث Save Project معناها احفظ المشروع
وأن Save project Diagrams احفظ خرائط المشروع
وأن Save project Narratives احفظ التفاصيل النصية للمشروع



مثال 2 :

في المخطط التالي لدينا حالة استخدام (اعرض الصنف) مستخدمة من قبل حالتي استخدام وهما (احذف الصنف) و (عدل الصنف).وهنا عرض الصنف ضرورية قبل تعديل أو إلغاء الصنف.



- **علاقة الامتداد Extend:**

علاقة الامتداد هي علاقة بين حالة استخدام وأخرى وتحدث عندما يكون لدينا أحد الأوضاع التالية:

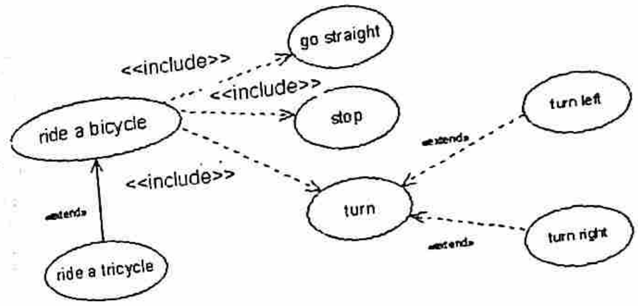
3. حالة استخدام رئيسية لديها حالات استخدام بديلة.
4. حالة استخدام إضافية(غير ملزمة أو ضرورية) يمكن إضافتها لحالة الاستخدام الرئيسية.

مثال 3 :

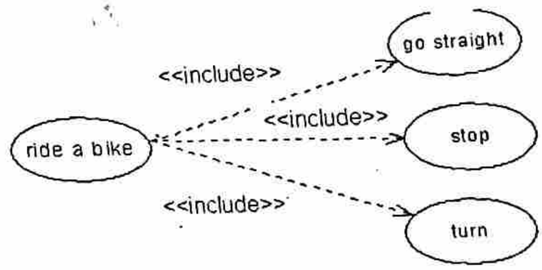
حالة الاستخدام التالية (قود الدراجة : **Ride a Bike**) :

لها 3 حالات فرعية:

- السير إلى الإمام Go Straight
- الوقوف Stop
- الدوران Turn

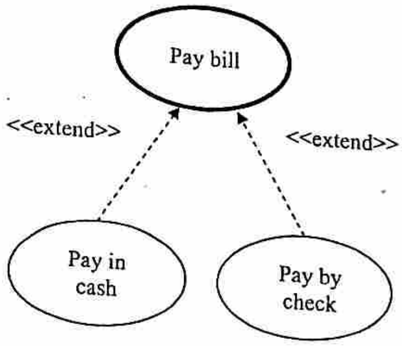


ويمكن تفصيل علاقة الامتداد كما يلي مع ملاحظة أن الدوران يكون على اليمين أو الشمال وان الذي يقود الدراجة العادية Bicycle (ذات العجلتين) يمكن أن يقود الدراجة ذات الثلاث عجلات Tricycle.



مثال 4 :
في حالة الاستخدام التالية (ادفع الفاتورة Pay Bill) في منظومة مبيعات يمكن أن يكون لها حالتي استخدام بديلة(اختيار) ولهذا نستخدم علاقة الامتداد كما يلي:

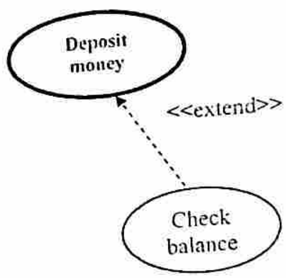
Pay in cash الدفع نقدا
Pay by a check الدفع بصك



ملاحظة: لاحظ اتجاه السهم في **Extend** عكس اتجاه السهم في **Include**

مثـــال 5 :

في حالة الاستخدام الرئيسية (ايداع النقود Deposit Money) في منظومة مصرفية يمكن أن يكون لها حالة استخدام إضافية (امتداد) مثل حالة الاستخدام (افحص الرصيد Check Balance) .



لاحظ أن حالة الاستخدام (فحص الرصيد) تعتبر غير ضرورية في حالة (إيداع نقود) بل هي حالة اختيارية، وهذا يعني وجود علاقة امتداد بينهما.بينما حالة الاستخدام(افحص الرصيد) تعتبر ضرورية في حالة (سحب صك) وهذا يعني وجود علاقة شمول بينهما.

مثـــال 6 :

نظام الحجز للخطوط الجوية

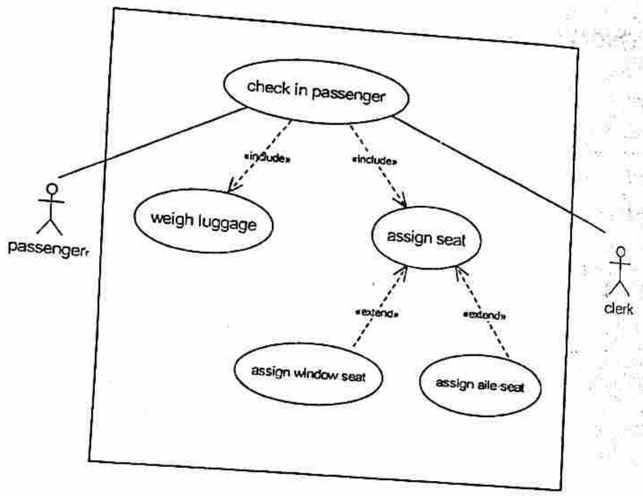
في هذا النظام لدينا حالات الاستخدام التالية:

إجراءات الصعود	Check-in passenger
زن الحقائب	Weigh luggage

خصص مقعد	Assign seat
خصص مقعد جانب النافذة	Assign window seat
خصص مقعد جانب الممر	Assign aisle seat

ولهم ممثلان وهما:

مسافر	Passenger
موظف الشباك	Clerk



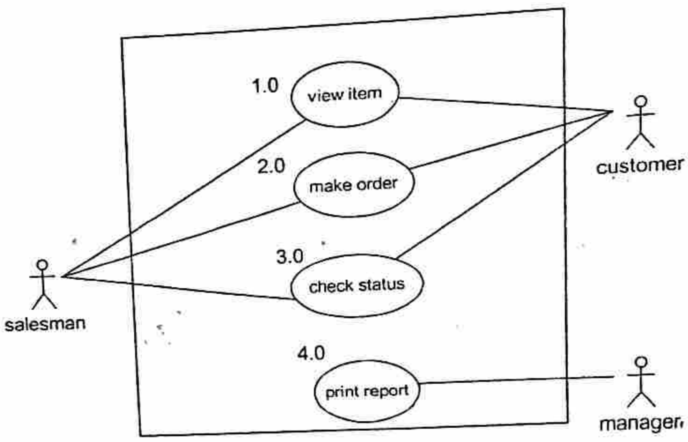
مثـــال 7 :

مخطط حالة الاستخدام لمنظومة معالجة طلبيات لمحل مبيعات يمكن تقسيمه إلى:

- مخطط حالة استخدام عام
- مخطط حالة استخدام مفصل

كما يلي :

والشكل التالي يبين مخطط حالة الاستخدام العام Overview UCD



ويمكن تجزئة حالة استخدام أعمل طلبيه Make Order حسب الاتي:

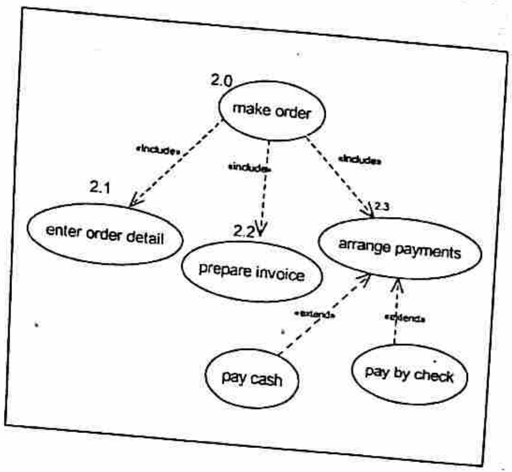
2.0 اعمل طلبية

2.1 ادخل تفاصيل الطلبية Enter Order details

2.2 جهز الفاتورة Prepare invoice

2.3 دفع قسط الفاتورة Arrange Payments

وهذا الترقيم يساعدنا عند توسيع المخطط، كما في المخطط التالي:



مـلاحظة : في المثال السابق افترضنا نوعين من الدفع:

- الدفع بصك **Pay by check**
- الدفع نقدا **Pay cash**

ولاحظ أيضا أن حالات الاستخدام الثلاثة الأولى ضرورية فاستخدمنا Include والحالتان (الدفع بصك أو الدفع نقدي) نختار أي منهما فنستخدم هنا Extend>

- **علاقـة التعميم : Generalization**

التعميم هو علاقة وراثة. وهي علاقة ممثل لممثل .

مثـال 8 :

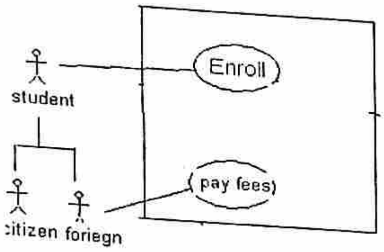
ارسم مخطط حالة الاستخدام لمنظومة تسجيل في جامعة حيث يوجد نوعان من الطلبة: طالب محلي لا يقوم بدفع رسوم، وطالب أجنبي يقوم بدفع الرسوم.

تقوم المنظومة بالعديد من الوظائف ولكن يركز المخطط هنا فقط على الوظائف التالية:

سجل في المقررات Enroll in courses

ادفع الرسوم Pay fees

ويبين المخطط التالي كيف نمثل هذه الحالة .



مثال على علاقة تعميم ممثل لممثل. في هذا المثال الممثل (الطالب) ينقسم إلى نوعين:

طالب مواطن Citizen student

طالب أجنبي Foreign student

لاحظ أن الطالب الأجنبي هو فقط الذي يدفع الرسوم.

7.7 سيناريـو حالة الاستخدام
Use case Scenario

سيناريو حالة الاستخدام عبارة عن سلسلة من الأحداث (الخطوات) اللازمة لأداء(حالة استخدام) معاملة أو وظيفة لمنظومة ما.

وهذا الوصف يأخذ شكل سيناريو يمثل ما يحدث عندما يتفاعل (ممثل أو فاعل) مع المنظومة.

قبل أن نعد السيناريو لحالة استخدام ما يجب علينا أولا تحدد حالات الاستخدام باستعمال بيان المسألة للمنظومة.

وبيان المسألة (أو تعريف المسألة **Problem Statement**) يكتب بلغة عادية أي لغة يفهمها الزبون كحوصلة للمقابلات والملاحظات التي تمت في موقع المستخدم لغرض التعرف على احتياجاته (متطلباته).

وبيان المسألة يجب أن يحدد جميع حالات الاستخدام للنظام كما نرى في الأمثلة التالية: .

مثـــال 1 :

وصف مسألة لنظام مصحة الآمال التخصصية :

في مصحة الآمال التخصصية يقوم المريض بالاتصال بالمصحة هاتفيا أو يأتي شخصيا للحصول على موعد مع الطبيب المختص الذي يريده، ويقوم موظف الاستقبال بالبحث عن وقت شاغر في كراسة المواعيد لتسجيل اسمه فيه. ويمكن للمريض إلغاء الموعد في أي وقت ولكن عليه أن يدفع الرسوم لموظف الخزينة مباشرة قبل إجراء الفحص من قبل الطبيب. وإذا لزم الأمر، عليه أن يدفع رسوما إضافية لبعض التحليل الطبية.

المطلوب تحديد الممثلين وحالات الاستخدام ثم رسم مخطط حالة الاستخدام UCD.

الحل:

من بيان المسألة أو التعريف المبين أعلاه ، يمكننا أن نحدد :

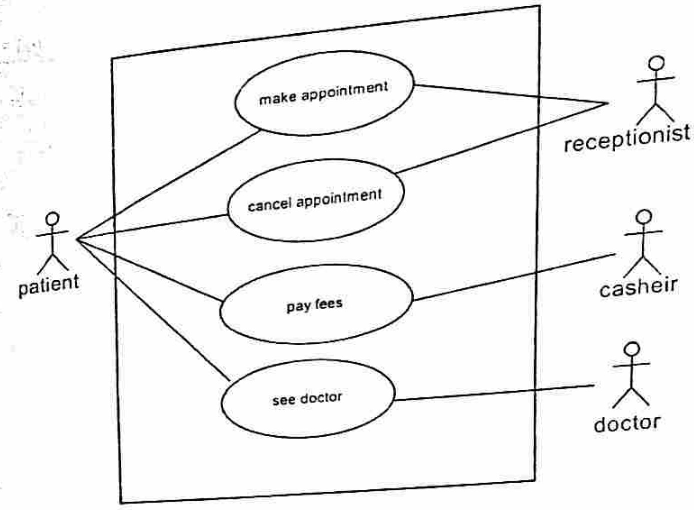
<u>الممثلين</u> (أي الذين يلعبون دورا في هذا النظام) :

PATIENT المريض
DOCTOR الطبيب
CASHIER موظف الخزينة
RECEPTIONIST موظف الاستقبال

<u>وحالات الاستخدام</u> هي :

Make appointment احجز موعد
Cancel appointment الغي موعد
Pay fees. ادفع الرسوم
See doctor ادخل للطبيب(أجراء الفحص)

ومخطط حالة الاستخدام لمنظومة المصحة تكون كما يلي:



مثـــال 2 :

تعريف المسالـــة يصف بصورة مبسطة منظومة التسجيل والدرجات

في كلية الأمل يقوم الطلاب بالتسجيل في المقررات، ويقوم المسجل بعملية التسجيل. أما أعضاء هيئة التدريس فيرصدون درجات الطلاب. ومن إحدى اختصاصات المسجل أن يعطي الطالب كشفا بجميع درجاته إذا طلب منه ذلك. حدد الممثلين وحالات الاستخدام

الحل:

المثلون في هذا النظام هم:

Student الطالب

Registrar المسجل

Instructor المعلم

حالات الاستخدام هي :

سجل في المقرر .

ارصد درجات الطالب .

امنح كشف درجات

مثـــال 3 :

ارسم مخطط حالة الاستخدامْ لمنظومة مكتبة كلية تقنية المعلومات في جامعة المجد مستخدما ملخص المسألة التالي Problem Statement :

ملـــخص المسألة لمنظومة مكتبة :

في مكتبة كلية العلوم بجامعة الضياء ، يقوم الطلاب والموظفون والمدرسون باستعارة الكتب، ويقوم موظف المكتبة بعملية الإعارة ، ويتحصل مدير المكتبة على تقارير يومية عن الكتب المستعارة والمرجعة.

ويقوم المشتركون بالاستفسار عن الكتب إما بواسطة عناوين الكتاب أو المؤلفين.

المطلوب:

1. حدد الممثلين في النظام
2. حدد حالات استخدام النظام.
3. ارسم مخطط حالات الاستخدام.

- **سيناريـــو حالة الاستخدام** Scenario

بعد تحديد كل حالات استخدام المنظومة، نبدأ بكتابة السيناريو الخاص لكل حالة استخدام على حده.

ويمكن تمثيل السيناريو نصيا على أنه مجموعة من الأفعال (الإحداث) والاستجابات بين الممثل والمنظومة.

فتوضع الأفعال الرئيسية (أي الناجحة) عادة في البداية وتسمى *بسيناريو الانسياب الأساسي* أما الأفعال الاستثنائية (الأخطاء مثلا) فتوضع بعدها وتسمى في هده الحالة *بالانسياب البديل* (أو سيناريو الانسياب الاستثنائي).

المثال التالي يمثل سيناريو حالة استخدام : *"ابحث عن كلمة"* في منظومة قاموس عربي/ انجليزي.

سيناريو الأنسيابات الأساسية (خطوات) هي :

1. المستخدم يدخل الكلمة باللغة الانجليزية
2. المستخدم ينقر بالفارة على زر البحث
3. المنظومة تنفد أمر البحث
4. المنظومة تعرض على الشاشة المعنى باللغة العربية

الانسيـــاب البديل :

إذا فشل البحث في الخطوة 3 ، يتم أعلام المستخدم بدلك من قبل المنظومة ويتم الرجوع مرة أخرى إلى الشاشة في الخطوة الأولى.

مـــلاحظات:

- يسمى سيناريو الانسياب الأساسي بالحالة الناجحة (عندما يكون البحث عن الكلمة ناجحا).
- يسمى سيناريو الانسياب الاستثنائي بالانسياب الممتد (عندما يكون البحث عن الكلمة غير ناجح).

كيف يتم تمثيل السيناريـــو ؟

يمكن تمثيل السيناريو بأحد الطرق الثلاثة الآتية :

1. **الفقــرة :** وهـي تصف الانسيابات الأساسية على شكل فقرة أو مجموعة فقرات تبين النشاطات (الإحداث) الخاصة بحالة الاستخدام.

مثـــال :

حالة الاستخدام : " أخذ موعد " في منظومة عيادة طبية

سيناريــو الانسيابات الأساسية :

يقوم المريض بالحضور شخصيا إلى العيادة أو بالاتصال هاتفيا لعمل موعد لمقابلة الطبيب. و يقوم موظف العيادة بتسجيل الموعد في كراسة المواعيد في حالة وجود مكان شاغر.

الأنسياب البديــل:

"اذا لم يجد الموظف مكانا شاغرا لهذا الموعد يتم توجيه المريض لأقرب عيادة"

2. قائمة العمود الواحد:

يبين المثال التالي سيناريو الانسيابات الاساسية لحالة الاستخدام "*تسجيل درجة امتحان*" من قبل المدرس لمنظومة معالجة درجات الامتحانات:

1. المدرس يدخل رقم الطالب
2. المنظومة تتحقق من أن الطا لب موجود في قاعدة البيانات
3. المنظومة تعرض أسم الطالب
4. يدخل المدرس رقم الامتحان

5. يدخل المدرس درجة الامتحان
6. المنظومة تدوّن درجة الامتحان ضمن سجل الطالب في قاعدة البيانات
7. المنظومة تخبر المدرس بأن الدرجة قد تم رصدها.

الانسيــاب البديل لحالة الاستخدام:

اذا لم يتم العثور على رقم الطالب في قاعدة البيانات ، فأن المنظومة سوف تعرض على الشاشة رسالة خطأ، ثم تعطي المدرس فرصة أخرى لإدخال الرقم الصحيح للطالب.

3. جــدول عمودين:

بدلا من عمود واحد يمكن أداء السيناريو بعمودين:

عمود يمثل الأفعال **Actions** التي يتم أداؤها من قبل الممثل

والآخر يمثل الإجابات **Responses**(الردود) من قبل المنظومة.

مثـــال :

هـذا السيناريو لحالة الاستخدام" *تسجيل درجة امتحان*" لمنظومة رصد درجات الامتحانات حيث يرصد المدرس درجة الامتحان.

والجدول التالي يبين هـذا السيناريو ذي العمودين . مع ملاحظة قراءة الجدول من الشمال إلى اليمين.

المنظومة (الردود)	المدرس (الافعال)
1. تتحقق من وجود رقم الطالب في قاعدة البيانات	1. يدخل رقم الطالب
2. تعرض أسم الطالب	
3. تقبل رقم الامتحان	2. يدخل رقم الامتحان
4. تخزن درجة الامتحان في قاعدة البيانات	
5. تفيد (أو تعلم) المدرس برسالة على الشاشة بأن الدرجة قد تم حفظها	3. يدخل درجة الامتحان

السيناريو البديل : اذا كان درجة الطالب غير موجودة في قاعدة البيانات ، يتم عرض رسالة خطاء على شاشة جهاز الحاسب ويتم توجيه المدرس بإدخال رقم صحيح للطالب.

- **الحالة السابقة والحالة اللاحقة :**

هل هناك عناصر أخرى إلى جانب الانسيابات الاساسية والانسيابات البديلة عند توثيق حالة الاستخدام ؟

الاجابة نعم . فحالة الاستخدام يمكن أن توصف بعناصر أخرى مثل:

- **الحالات السابقة** Pre condition : الشروط التي يجب تلبيتها قبل بدء حالة الاستخدام.
- **الحالات اللاحقة** Post condition : الشروط التي تحدث عند انتهاء حالة الاستخدام.

مثال :

"اسحب نقدا" من آلة السحب الذاتي بمصرف الأمان.

ماهي الحالات السابقة واللاحقة لحالة استخدام ؟

قبل أن يشرع الزبون في سحب نقد من الآلة يجب أن تتوفر الشروط التالية:

- تحميل نقد كافي في الآلة
- عرض ما يفيد استعداد الآلة لقبول بطاقة السحب الذاتي ويتم ذلك بعرض شعار المصرف مع طلب من الزبون بإدخال بطاقة السحب الذاتي.

وتسمى الشروط المذكورة بالحالات السابقة.

وبعد انتهاء الزبون من تنفيذ المعاملة تقوم المنظومة ب:

- عرض ما يفيد بأن المعاملة السابقة قد تمت بنجاح

- عرض ما يفيد استعداد الآلة لقبول بطاقة السحب الذاتي.

7.8 أسئلة للمراجعة

1 - قارن بين :

- Actor وبين Used Case
- Extend وبين Include
- Object وبين Class
- Attribute وبين Operation
- Operation وبين Method

2 - ما معنى المصطلحات التالية؟ هات مثالا لكل منها.

- Association
- Aggregation
- Composition
- Generalization
- Inheritance

3- قارن بين التحليل والتصميم الشيئي والهيكلي؟

4 - اذكر مثالا لمخطط حالة استعمال مبينا علاقة امتداد وعلاقة شمول.

5 - اذكر 3 لغات شيئية؟

6 - أيهما تعتقد تم إعداده قبل الآخر:

البرمجة الشيئية أو التصميم الشيئي؟

7 - اكتب T في حالة صح و F في حالة خطأ أمام الجمل التالية:

- يحتوي مخطط حالة الاستخدام على حالات استخدام وممثلين.( )
- لا يمكن استعمال مخطط حالة الاستخدام للنظم الهيكلية بل للنظم الكائنية فقط. ( )
- يستعمل مخطط حالة الاستخدام لتوضيح المتطلبات الوظيفية للمنظومة ( )
- تستعمل لغة UML لتحليل وتصميم النظم باستخدام المفاهيم الشيئية ( )
- مخطط حالة الاستخدام أداة جيدة للتواصل مع الزبون لغرض الحصول على متطلباته. ( )
- مخطط حالة الاستخدام يبين ما يقوم به النظام لا كيف يقوم به.( )
- التغليف هو عملية إخفاء الوظائف الداخلية عن الآخرين. ( )
- الوراثة تعني أن الوظائف نحصل عليها من فصيلة الأب( ).

8 - ما هي أهم مكونات مخطط حالة الاستخدام؟

9- ما هو الفرق بين الممثل والمستخدم؟

10- أرسم مخطط UCD لمنظومة مخازن.

11- اكتب سيناريو حالة استخدام "استعارة كتاب" لمنظومة مكتبة.

12- أذكر أمثلة من عندك :

- حالة استخدام
- ممثل
- علاقة امتداد Extend
- علاقة شمول Include

13- أذكر الطرق الثلاثة لكتابة السيناريو؟ أيهم تفضل؟ ولماذا؟

8

تصميم النظم

Systems Design

Chapter Eight

8.1 مقدمـــة

يمكن تعريف التصميم بانه "عملية تمثيل للحل باستخدام مخططات وجداول وواجهة مستخدم وخوارزميات". ومرحلة التصميم لها علاقة بمكونات نظام المعلومات. والمعلوم ان المكونات الاساسية لنظام المعلومات هي : البرمجيات والعتاد والبيانات والعنصر البشري. إن مرحلة التصميم تتكون من مجموعة نشاطات تتعلق بالمكونات سالفة الذكر.

والسؤال هنا ما هو اول نشاط يجب ان نبدا به في مرحلة التصميم؟

- ان اول نشاط يجب ان نبدا به يتعلق بالعنصر البشري (المستخدم) والذي يعتبر العنصر الاساسي لاي منظومة نرغب في اعدادها . ويسمى هذا النشاط " تصميم واجهة المستخدم" User Interface او التفاعل والتواصل بين المنظومة والمستخدم والمتمثل في تصميم شاشات الادخال والاخراج وكذلك تصميم التقارير المنبثقة من المنظومة .

اما النشاط الثاني فهو التصميم المعماري ثم تصميم البيانات وفي النهاية تصميم الاجزاء البرمجية او اعداد الخوارزميات لكل الاجزاء البرمجية.

هناك ملاحظة مهمة وهي ان النشاطات المتعلقة بالتصميم مرتبطة ببعضها البعض وأن أي تغيير في أي نشاط يتطلب تغيير في

النشاط الاخر فمثلا تغيير عناصر شاشة ادخال قد يتطلب تغيير في تصميم البيانات والخورازميات المتعلقة به وهكذا.

ان الهدف الأساسي من مرحلة التصميم هو اعداد منظومة خالية من الاخطاء وسهلة الصيانة والاستخدام وتلبي في النهاية متطلبات المستخدم . لهذا يجب ان نهتم بعدة اعتبارات تتعلق بالمستخدم والبيانات والاجراءات (المعالجة) والتي منها مشاركة المستخدم في تصميم الشاشات والتقارير وتجزئة المنظومة واستخدام المخططات الرسومية والجداول كأدوات تصميم.

8.2 التصميم المعماري Architectural Design

ويقصد بالتصميم المعمارى تجزئة المنظومة المراد إعدادها إلى مكونات (أجزاء برمجية) في شكل هرمي ، والغرض هو تبسيط المنظومة المعقدة لأجل سهولة تصميم خوارزمياتها و تسهيل عملية توزيعها على المبرمجين.

في هذا الباب سوف نستعرض بعض الأدوات الشائعة والمستخدمة في التصميم المعماري **Architectural Design** والتي منها المخطط الهيكلي**Structure chart** ومخطط ورنيير **Warnier-Orr** .

ثم نتطرق إلى بعض المفاهيم العامة في التصميم ومعايير التصميم الجيد.

8.3 المخطط الهيكلي Structure Chart

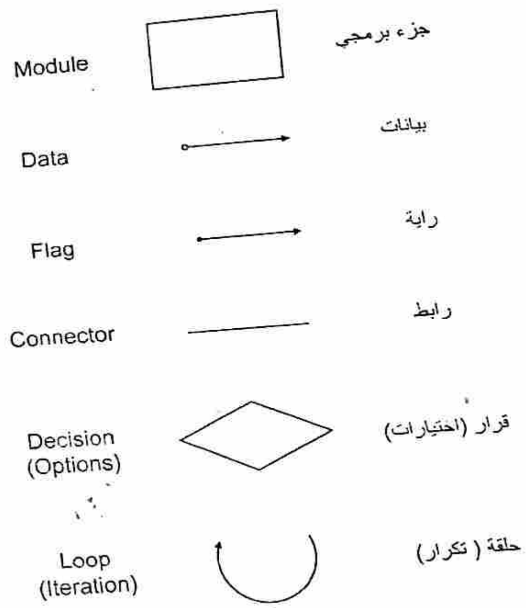
المخطط الهيكلي هو أداة تصميم رسومية توضح تركيبة (هيكلية) المنظومة في شكل هرمي.

ويبين هذا المخطط العناصر الآتية :

- الهيكلية الشجرية أو الهرمية الكاملة للمنظومة كأجزاء برمجية(**Modules**) .
- العلاقة بين هذه الأجزاء **Relationship** .
- البينيات (**Interfaces**) وهي البيانات المررة بين هذه الأجزاء البرمجية .

مع ملاحظة أن هناك جزءا برمجيا للتحكم **Control module** سوّاق في أعلى مستوى وهو الذي يقود الأجزاء البرمجية الأخرى .

ويبين الشكل (8.2.1) الرموز التي عادة تستخدم في المخطط الهيكلي.



الشكل (8.3.1) رموز المخطط الهيكلي

ملاحظة: الرموز ذات العلاقة بتمرير البيانات ورايات التحكم تسمى عادة البينيّات **Interfaces** بين الأجزاء البرمجية.

قواعد استعمال المخطط الهيكلي :

1. الأجزاء ذات المستوى المنخفض (في أسفل مستوى) تقوم بالمعالجة بينما الأجزاء ذات المستوى العالي تقوم بوظيفة التحكم.
2. استعمل (فعل أمر + مفعول به) مثل (احسب المرتب) لوصف العملية التي يقوم بها جزء برمجي معين .وهنا نستخدم الاسم والفعل دون استخدام الصفة والظرف وغيرها.

و يمكن أن نستعمل الأفعال التالية لهذا الغرض :

Get	احصل على
Read	اقرأ
Validate	حقق
Compute	احسب
Check	افحص
Write	اكتب
Print	اطبع
Format	شكّل
Determine	قرّر

3- يكون للأجزاء البرمجية ذات المستوى العالي أفعال عامة بلغة المستخدم تبين العملية ذات المستوى العالي مثل:

Update	حدّث
Edit	عدّل
Process	عالج
Prepare	جهز

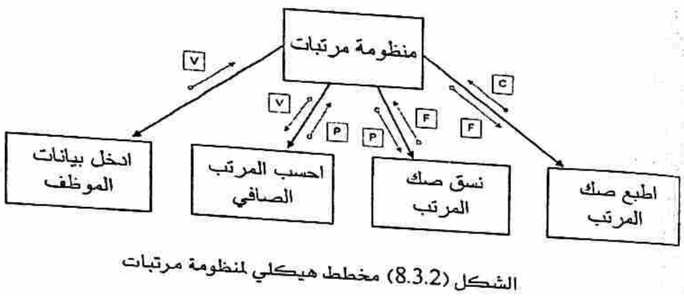
ولكن الأجزاء ذات المستوى المنخفض يكون لها أفعال محددة للإشارة للمعالجة ذات المستوى المنخفض.
وهذه الأفعال تكون عادة مصطلحات حاسوب مثل :

Format	شكّل
Display	اعرض
Print	اطبع
Compute	احسب
Verify	حقق
Retrieve	استرجع

مثـــال :

دعنا نفحص المخطط الهيكلي التالي لمنظومة المرتبات كما مبين بالشكل (8.3.2). نلاحظ في هذا الشكل أن منظومة المرتبات تتكون من الأجزاء البرمجية التالية :

- ادخل بيانات الموظف
- احسب صافي المرتب
- نسق صك وقائمة المرتبات
- أطبع صك وقائمة المرتبات



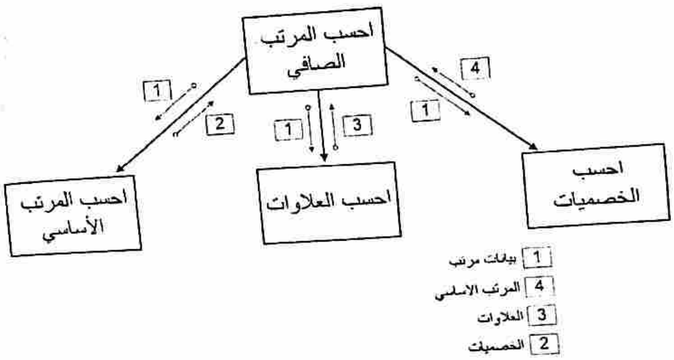
الشكل (8.3.2) مخطط هيكلي لمنظومة مرتبات

معاني الرموز في المخطط كالآتي :

V	بيانات المرتب
P	معلومات لصك وقائمة المرتب
C	راية انتهاء العمل
F	صك وقائمة مرتبات منسقة

ويمكن تقسيم الجزء "احسب المرتب الصافي" إلى أجزاء فرعية (كما مبين بالشكل 8.2.3) وهي :

- احسب المرتب الأساسي
- احسب العلاوة
- احسب الخصميات



الشكل (8.3.3) أجزاء فرعية للجزء "احسب المرتب الصافي"

1 = معلومات الموظف
=2 الراتب الأساسي
3 = العلاوات
4 = الخصميات

في الشكل (8.3.3) مربعات تنفذ من اليسار إلى اليمين تتابعيا .

ويوصف هذا النظام عادة بأنه نظام تحوّلي أو منظومة تحولية.Transform Centered system أي أن الإدخال من الشمال ثم المعالجة في المركز ثم الإخراج في جهة اليمين.

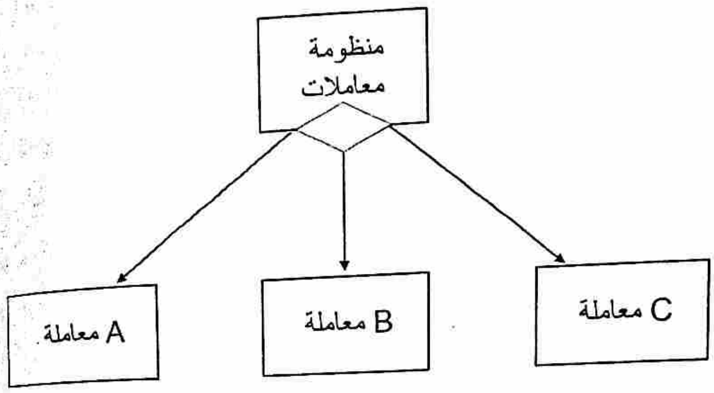
وفي التطبيقات التجارية عادة ما يستعمل نوع واحد فقط من المعاملات في الوقت الواحد. ويوصف هذا النظام بأنه مركز على المعاملة أو منظومة المعاملات Transaction-Centered System

ومن أنواعه :

Inventory Systems	منظومة المخزون
Banking Systems	منظومة المصارف
Medical System	منظومة طبية
Air-line Reservation Systems	منظومة الحجز في الخطوط الجوية
Hotel systems	منظومة الفنادق
Sales systems	منظومة المبيعات

ولتوضيح هذا النوع (منظومة المعاملات) ندرس الشكل التالي:

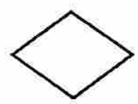
نفرض أن Trans اختصار لكلمة Transaction بمعنى معاملة (أي وظيفة تؤديها المنظومة).



الشكل (8.3.4) منظومة المعاملات

نلاحظ أن الجزء البرمجي المتحكم **Driver** أو السوّاق في الشكل (8.2.4) يقوم بتحويل التحكم لإحدى عمليات المعاملة بناء على المعلومات الواردة إلى هذا الجزء السواق.

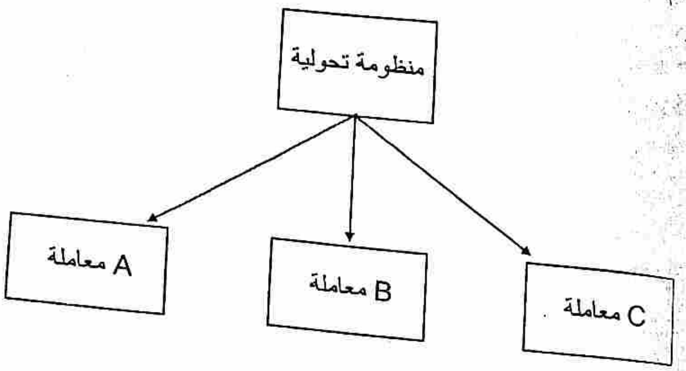
ولاحظ أن المخطط المرتكز على المعاملة يستخدم دائما الرمز التالي:



هذا الرمز هو أداة ضبط التحكم حيث تتفرع منه عدة معاملات ولكن يتم تنفيذ واحدة فقط اعتمادا على نوع المعاملة المطلوبة.

أما المنظومة التحولية فهي كالمبينة في الشكل التالي:

يتم هنا تنفيذ الأجزاء البرمجية كلها وتباعا المعاملة **A** ثم المعاملة **B** ثم المعاملة **C**.

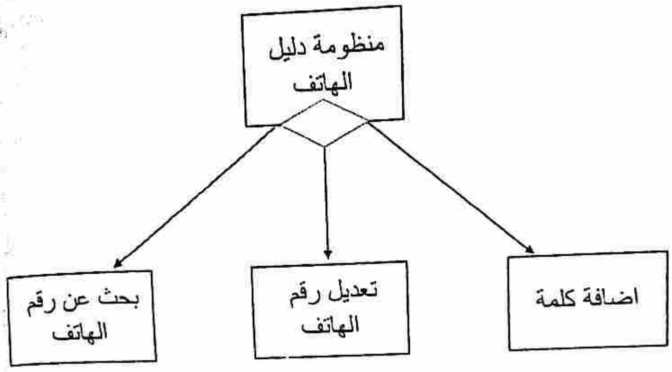


الشكل (8.3.5) منظومة تحولية

مثال: ارسم المخطط الهيكلي لمنظومة قاموس انجليزي/ عربي **Dictionary** لأداء للوظائف التالية :

- البحث عن معنى كلمة
- إضافة كلمة جديدة
- تغيير معنى كلمة

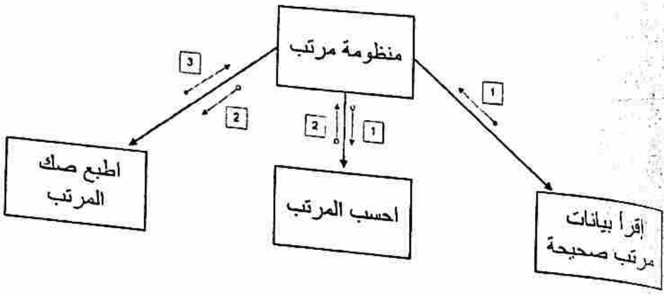
حيث يبين الشكل التالي المخطط المطلوب.



الشكل (8.3.6) منظومة قاموس انجليزي/عربي

والآن دعنا نفحص المنظومة التحولية كما هو مبين بالشكل 8.3.7 والذي يمثل نظام دفع الرواتب بالصك والمتكون من الأجزاء التالية :

- ادخل بيانات صحيحة
- احسب المرتب
- أطبع صك المرتب



الشكل (8.3.7) منظومة مرتبات تحوّلية

الـــــرموز :

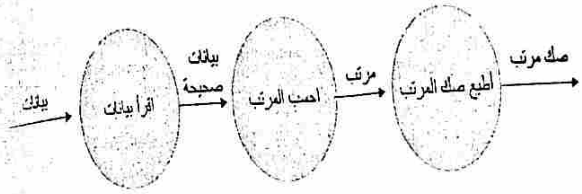
1 : بيانات صحيحة

2 : المرتب

3 : راية النهاية

والمنظومة في الشكل 8.2.7 يوضح مثالا لمنظومة تحولية حيث تتبع العمليات نفس المسار.وهنا يتم التنفيذ من اليمين الى الشمال، أي قراءة بيانات المرتب أولا ثم حساب المرتب الصافي وفي النهاية طباعة الصك والقائمة.

ويمكن توضيح ذلك أكثر بالشكل 8.2.8 وهو مخطط انسياب البيانات **DFD** أي **Data Flow Diagram** .



الشكل (8.3.8) لمنظومة تحولية **DFD of Transform-Centered System**

في هذا الشكل نرى أن البيانات تتحول من مرحلة إلى أخرى، حيث نبدأ بالحصول على بيانات صحيحة ثم نحسب قيمة المرتب، ثم أخيرا نطبع صك المرتب.

من دراسة الشكل **DFD** يستطيع المصمم أن يقرر التركيبة الفوقية المناسبة (التحويل أو المعاملة).

مثـــال :

حوّل منظومة الرواتب المبين بالشكل 4.3.8 من **DFD** إلى المخطط الهيكلي .

ملاحظة : عملية تحويل مخطط انسياب البيانات إلى المخطط الهيكلي تسمى بالتحليل التحويلي **Transform Analysis**.

الحل : لتحويل مخطط انسياب البيانات إلى مخطط هيكلي اتبع الخطوات التالية :

1. كوّن جزء تحكم (سواقة) تحت اسم معين وليكن باسم (نظام دفع الرواتب) في المخطط الهيكلي.
2. كل عملية **Process** في **DFD** يتم تغييرها إلى جزء برمجي (**Module**) في المخطط الهيكلي بنفس الاسم.
3. اربط هذه الأجزاء البرمجية بالسواقة مستخدما الوصلة.
4. انسيابات البيانات **Data flows** في مخطط **DFD** يجب أن تتحول إلى رموز بيانات أو رايات في المخطط الهيكلي .

ملاحظة : المخطط الهيكلي مبين في الشكل (8.3.7)

مثـــال :

افترض أننا نريد طباعة بيانات جميع الزبائن لمصنع الثلاجات على الشاشة . ارسم المخطط الهيكلي لهذا المنظومة.

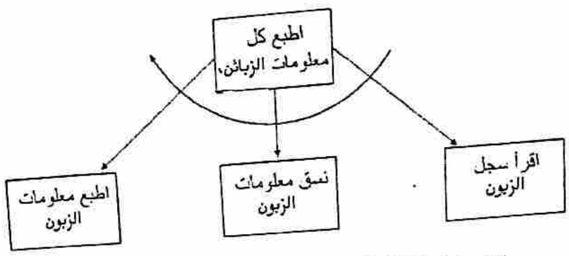
الحل:

يجب هنا استخدام رمز التكرار :



وذلك لغرض بيان أن العملية يجب أن تتكرر عددا من المرات كما يلزم.

المخطط المطلوب مبين بالشكل التالي :



الشكل (8.3.9) توضيح التكرار في المخطط الهيكلي

8,4 تصميم واجهة المستخدم User Interface Design

يتعلق هذا النوع من التصميم بالتعامل والتواصل بين المنظومة ومستخدميها.

فقبل تصميم واجهة المستخدم والمتعلقة بتصميم الشاشات والتقارير يجب على محلل النظم ان يدرس إمكانيات ورغبات المستخدم عند التعامل مع المنظومة.

ويتعلق تصميم الواجهة بتصميم المدخلات والمخرجات على الشاشة والتقارير على الورق.

ويجب أخذ الآتي في الاعتبار:

1. التركيز على تلبية الأهداف الأساسية المتوخاة من المنظومة المراد إعدادها.
2. جعل هذه الواجهة سهلة التعلم و الفهم والعمل بها **User Friendly**.
3. جعل الواجهة تؤدي جميع الوظائف المنوطة بها.
4. تستخدم الواجهة القوائم والرسومات (الأيقونات) قدر المستطاع.
5. جعل الواجهة تتسم بالوضوح وعدم الالتباس في المصطلحات.
6. توفير المساعدة **Help** في حالة عدم القدرة على أداء أي وظيفة أو في حالة حصول ارتباك في التعامل مع المنظومة..
7. جعل الواجهة تتعامل مع الأخطاء وتقلل حدوثها.
8. استخدام العروض التجريبية **Prototypes** مع الزبون لأخذ ملاحظاته وتنفيذ الصالح منها
9. التقليل من مشاكل إدخال البيانات.
10. استخدام مصطلحات بيئة عمل الزبون.
11. تصميم شاشات جذابة للمستخدم.
12. تقليل حجم البيانات المدخلة للمنظومة.
13. إدخال البيانات الضرورية فقط.
14. استخدام التشفير بقدر الإمكان فمثلا يمكن إدخال أو اختيار (**M** بدلا من **Male**).

15. التقليل من الإدخال لأن العنصر البشرى المتمثل في المستخدم عرضة أكثر للأخطاء والحل هو عملية عرض قوائم يتم اختيار البنود المرغوبة باستخدام الفأرة.

طرق التفاعل بين المستخدم والحاسوب :

التفاعل بين البشر والحاسوب باستخدام الشاشات يمكن أن يتم بإحدى الطرق التالية :

النص	Text
القائمة	Menu
الأيقونات	Icons

كيف تصمم واجهـة مستخدم رسومية جيدة ؟

معظم التطبيقات البرمجية (أو المنظومات) الحديثة تتفاعل مع المستخدم عن طريق ما يعرف باسم **GUI** (واجهة المستخدم الرسومية) وهي نوع من أنواع التعامل مع المستخدم.

وباستخدام **GUI** يتم التواصل بين الناس والحاسوب عن طريق تمثيل الكائنات رسوميا (الأيقونات) واستخدام صور الأزرار والنوافذ والقوائم الأفقية والعمودية التي تظهر على شاشة جهاز الحاسوب.

واستخدام المؤشرات (مثل الفأرة) يصبح ضروريا عند استعمال **GUI**.

لقد أصبحت التطبيقات ذات الوسائط المتعددة شيئا فشيئا شائعة الاستعمال .

إن طريقة **GUI** وباستخدام الرسوم، والفيديو، والصوت، والصور المتحركة (كما في نظام ويندوز) بدأت تحل محل الطرق القديمة ذات الأوامر النصية (كما في نظام دوس).

مزايـــا واجهة المستخدم الرسومية :

1. سهلة التعلم والاستخدام بعد دورة قصيرة.
2. يمكن أن تكون للمستخدم نوافذ متعددة على شاشة واحدة حتى يستطيع أن يتنقل من نافذة إلى أخرى.
3. إن إضافة الصورة والألوان تجعل من المنظومات أكثر جاذبية ومتعة عند الاستخدام.

إرشادات هامة عند تصميم واجهة المستخدم :

1. استعمل خصائص متنوعة من **GUI** مثل الأيقونات والقوائم والأزرار وصناديق الاختيار وشريط السحب وشريط التقدم.
2. يجب تمكين المستخدمين من التراجع عن الأمر أو إعادته.
3. وفر المساعدة الفورية (خط مباشر) ولكن قلل من الحاجة إليها.

4. استخدم الصوت والحركة والفيديو والنص المترابط كلما أمكن ذلك.
5. خذ في الاعتبار نوع المستخدم للمنظومة . مثلا إذا كان المستخدم غير ملم بالحاسوب فيجب مراعاة قدراته ونفسيته.

مثـــال : منظومة مخازن

نفترض أن المطلوب إعداد منظومة مخزون تقوم بالوظائف التالية :

البداية تكون تصميم واجهة المستخدم لعرضها على الزبون وأخذ ملاحظاته التي ستسهم إلى حد كبير في قبول المنظومة.

المعامـــلات Transactions

1. إضافة صنف **Add item**
2. تعديل صنف **Edit item**
3. إلغاء صنف **Delete item**
4. استفسار **Query**
5. تقارير **Reports**

5.1 تقرير عن أصناف **Item report**
5.2 تقرير عام **general report**

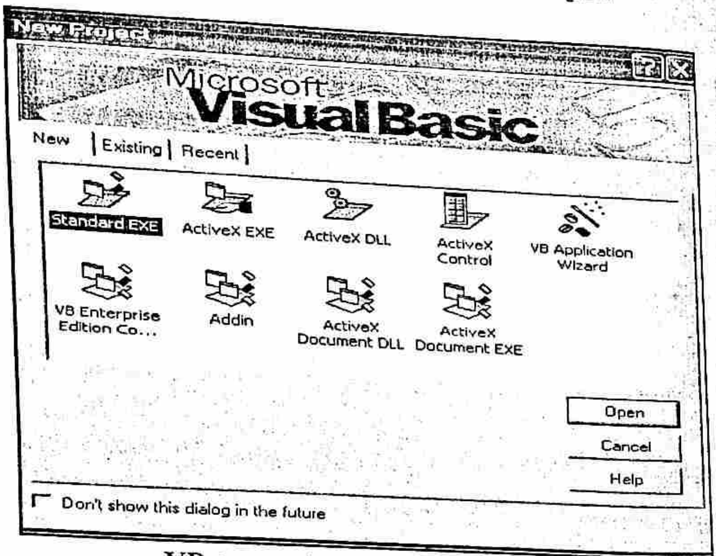
نلاحظ هنا أن المنظومة تملك وظيفتين هما: المعاملات والتقارير.

في هذا المثال نستخدم بيسك المرئي 6.0 في تصميم واجهة المستخدم الرسومية.

وفي الواقع يمكننا استخدام أي لغة برمجة مرئية لهذا الغرض.

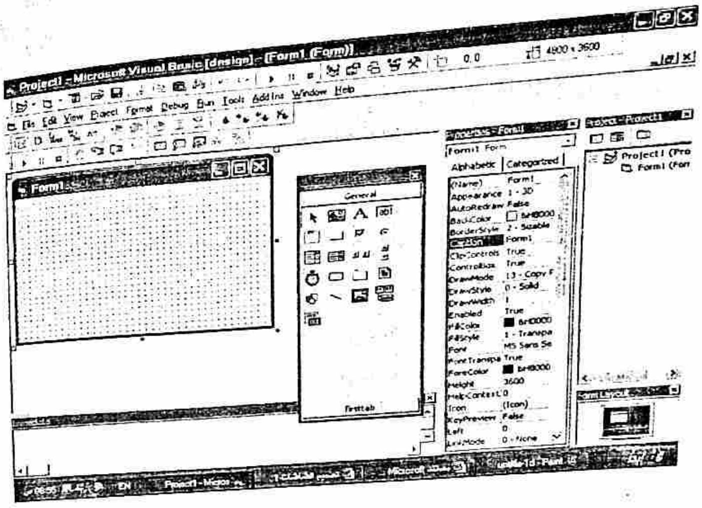
وفيما يلي الخطوات التي يمكن أن نحصل على المطلوب باستخدام لغة بيسك المرئي **Visual Basic** أو باختصار **VB**.

1. من قائمة البرنامج انتق برنامج **VB** لتحصل على الشاشة التالية:



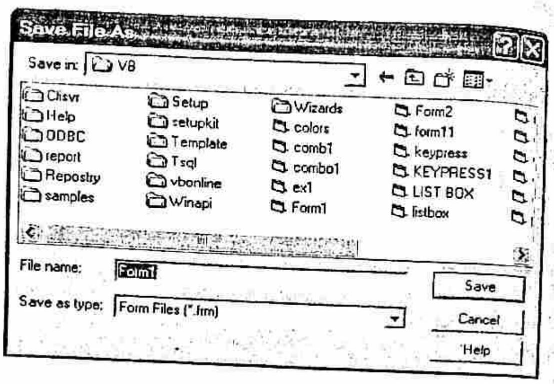
الشكل (**8.4.1**) الشاشة الأولى في بيئة**VB**

2. انتق **Standard.exe** للحصول على الشاشة التالية المسماة بيئة الإعداد المتكامل **IDE** .



الشكل **(8.4.2)** بيئة الإعداد المتكامل

من القائمة الرئيسية انتق **File** ثم **New project** للحصول على مربع الحوار التالي:

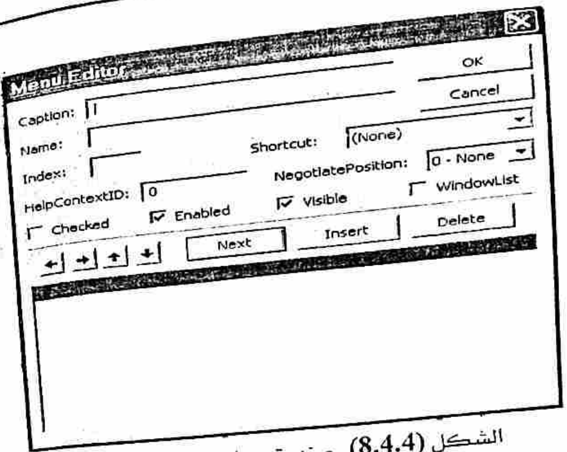


الشكل 8.4.3 صندوق حوار (حفظ نموذج)

المطلوب هنا تخزين المشروع والنموذج تحت اسم مناسب.

اكتب مثلا example1 بدلا من form1 و بدلا من project1.

الآن نستطيع تصميم القائمة المطلوبة باستخدام menu editor محرر القائمة التي ننتقيها من القائمة العمودية للأدوات Tools لنحصل على ما يلي:



الشكل **(8.4.4)** صندوق حوار لمحرر القائمة

هنا يجب أن نكتب عنوانا في **Caption** واسما **Name** لأول بند في القائمة ويمكنك كتابة (معاملات) بالنسبة للعنوان و **Trans** للاسم ثم اضغط على زر **(next)** لإضافة عنصر آخر للقائمة الرئيسية وهو (تقارير) في هذه الحالة.

عندما ننفذ المشروع نتحصل على قائمة أفقية تحتوي على :

Transactions معاملات **Reports** تقارير

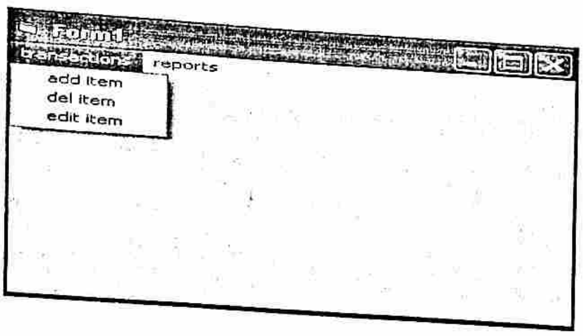
ولكن هذا لايكفي . نحتاج إلى قائمة عمودية تظهر عندما يتم النقر على المعاملات أو التقارير. لأداء ذلك ، افتح محرر القائمة وأضف العنوان (أضف صنف add item) بين (المعاملات) و(التقارير) مستخدما زر insert. علما بأن النقاط الواردة قبل

(أضف صنف add item) تبين أنه عبارة عن عنصر متفرع من (المعاملات).

وللحصول على هذه النقاط استعمل الزر ← الموجود في محرر القائمة.

ويمكن الحصول على العناصر العمودية الأخرى للقائمة بنفس الطريقة.

والآن إذا نفذنا (example1) سيعرض علينا نموذجا بقائمة أفقية تحتوي على (المعاملات) و (التقارير).



الشكل **(8.4.5)** قائمة أفقية وعمودية

3. نريد الآن أن نبين كيف نستخدم النموذج لإدخال البيانات.

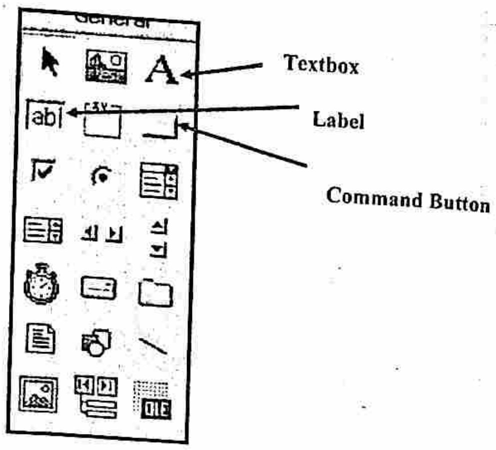
دعنا ندرس ماذا يجب أن نفعل عند النقر على (أضف صنف).

افترض أننا نريد إدخال البيانات التالية عند إضافة صنف :

- رقم الصنف
- وصف الصنف
- سعر الصنف
- كمية الصنف

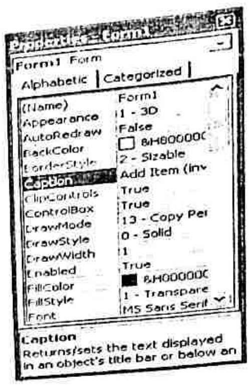
وبعد إدخال البيانات نعطي المستخدم الاختيار بين حفظ البيانات أو تعديلها أو الخروج. لعمل ذلك في بيسك المرئي نصمم نموذجا باستخدام أدوات التحكم **Label** و **Text Box** الموجودة بصندوق الأدوات.

ولتمكين المستخدم (في هذا النموذج) من حفظ البيانات أو تعديلها أو الخروج نستعمل زر الأمر .



الشكل 8.4.6 صندوق الأدوات

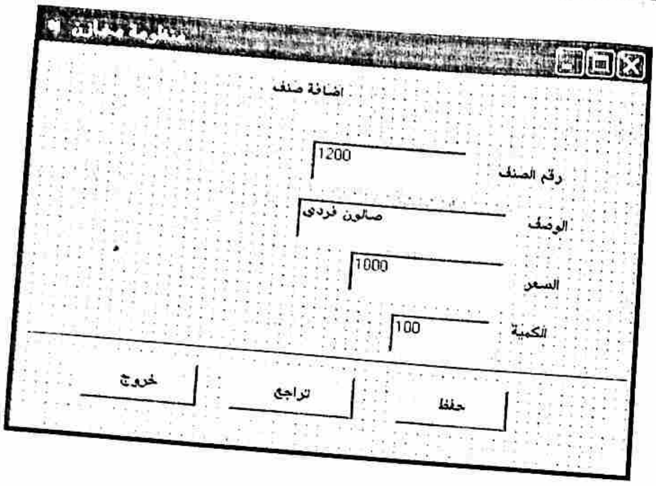
لكتابة العناوين (مثل أضف صنف Add item) استخدم الخاصية caption من شاشة الخصائص·انظر الشكل (8.4.7)



الشكل 8.4.7 نافذة الخصائص

وبنفس الطريقة يمكن كتابة العنوان المناسب في النموذج باستخدام نافذة الخصائص.

والآن عند تنفيذ المشروع والنقر على (أضف صنف) يجب أن نحصل على النموذج المبين بالشكل (8.4.8)



الشكل 8.4.8 نموذج إدخال البيانات صنف

أنـــواع التقارير:

1. تقارير تفصيلية **Detailed reports**
2. تقارير مختصرة **Summary reports** وهي تخص المدراء للمساعدة في اتخاذ القرار
3. كشوفات أو قوائم عند الطلب تتعلق بجزء من البيانات

8.5 تصميم البيانــات Data Design

يعتبر تصميم البيانات **Data Design** من النشاطات المهمة عند إعداد المنظومات. ويبدأ تصميم البيانات من مرحلة التحليل حيث يتم التعرف على البيانات الخاصة بالمنظومة ومن ثمّ تنظيمها وهيكلتها حتى تصبح سهلة التخزين والاسترجاع والتغيير.

والهدف الأساسي من تصميم البيانات هو تحويل البيانات التي تمّ إعدادها في مرحلة التحليل إلى هياكل بيانات ليتم استخدمها في مرحلة البرمجة.

وعند تصميم البيانات يجب تجهيز الآتي :

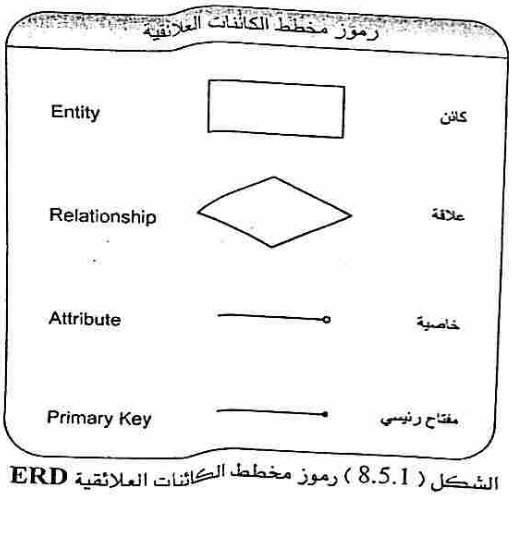
- قاموس البيانات **Data Dictionary** للتعرف على كل عناصر البيانات الخاصة بهذه البيانات وخصائصها.
- الكائنات والعلاقة بين هذه الكائنات.
- الخصائص الخاصة بكل كائن.
- مخطط الكائنات العلائقية **ERD**
- الجداول العلائقية المتحولة من مخطط الكائنات العلائقية .

في البنود التالية سنتعرض إلى مخطط الكائنات العلائقية وقاموس البيانات وهما الأداتان الرأسيتان في تصميم البيانات.

مخطط الكائنات العلائقيـــة ERD

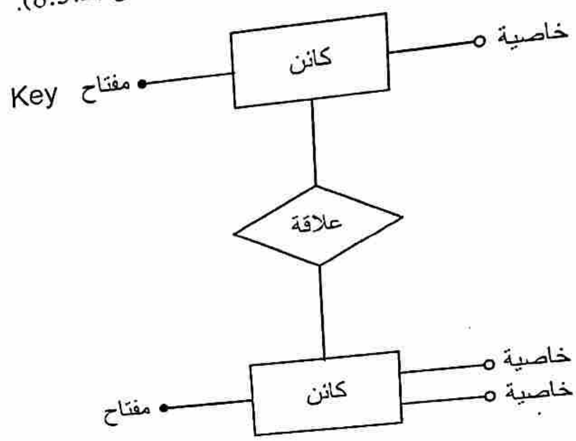
يعتبر بيتر تشن هو أول من أعدّ أداة التصميم **Entity Relationship Diagram -(ERD)** ، وكان ذلك في أواخر السبعينيات لغرض تصميم نظم قواعد البيانات العلائقية.

وأصبحت هذه الأداة ذات أهمية في تصميم هياكل قواعد البيانات. والهدف الرئيسي من هذا المخطط هو توضيح عناصر البيانات (الكائنات) وخصائصها والعلاقات بين هذه الكائنات.والشكل التالي يبين مخطط الكائنات العلائقية :-



الشكل (8.5.1) رموز مخطط الكائنات العلائقية **ERD**

في مخطط الكائنات العلائقية يمكن أن يوجد كائنان مرتبطان بعلاقة مع خاصية ومفتاح رئيسي كما في الشكل (8.5.2).



الشكل (8.5.2) مخطط الكائنات العلائقية

مثـــال :

في الكائن (**Customer** أي الزبون) يوجد أربعة خصائص (حقول) ومفتاح رئيسي واحد:

رقم الزبون (مفتاح) **Primary key : Customer-id**

اسم الزبون (الحقل الأول) **Attribute 1 : Customer-name**

العنوان (الحقل الثاني) **Attribute 2 : Address**

الحالة {جديد / قديم} (الحقل الثالث) **Attribute 3 : Status**

الرصيد (الحقل الرابع) **Attribute 4 : Balance**

:

مثـــال

يعتبر الكتاب والزبون كائنين. والعلاقة بينهما كما يلي :

- الزبون يستعير كتابا
- الزبون يستفسر عن كتابا
- الزبون يعيد كتابا

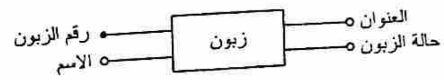
على سبيل المثال يمكننا توضيح العلاقة كما يلي:

الكائن زبون والخصائص التابعة له

الكائن كتاب والخصائص التابعة له

والعلاقة بين الكائن زبون والكائن كتاب

ويمكن توضيحها بالمخطط كما يلي :





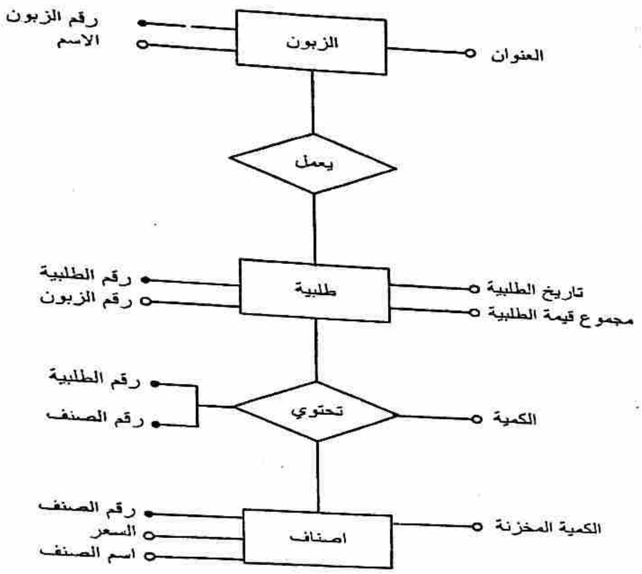


هذا المخطط يساعدنا في الحصول على أساس لتكوين جداول قاعدة البيانات ، حيث يمكن تكوين جدول لكل كائن.ويمكن في بعض الحالات تكوين جدول للعلاقة كما يتضح في الأمثلة اللاحقة.

مثال : ارسم مخطط كائنات علائقية لمنظومة مبيعات حيث يقوم الزبون بعمل طلبية تحتوي على أصناف متنوعة .

الحـل :

المخطط التالي يبين بطريقة مبسطة الكائنات في هذه المنظومة والعلاقة بينها ، مع توضيح مجموعة محددة من الخصائص حسب الشكل التالي :



الشكل(8.5.3) مخطط كائنات علائقية لمنظومة مبيعات

مثـــال :

ارسم مخطط الكائنات العلائقي لشركة **تعمير** تعمل على مشاريع متعددة في مجال البناء والتشييد،

الحـــل:

الكائن الرئيسي في هذه المنظومة هو (مشروع) وله خصائصه كالتالي:

Project number رقم المشروع

1. **Project title** اسم المشروع
2. **Budget** الميزانية
3. **Starting date** تاريخ البداية
4. **End date** تاريخ النهاية
5. **Type** نوع المشروع
6. **Specialty** تخصص الشركة
7. **Issue-date** تاريخ إصدار التقرير

أمّا العلاقات فهي كالآتي :

- الشركة لها مشاريع **(Company owns projects)**
- المشروع يستخدم تقارير **(Project uses report)**

وعليه يكون مخطط الكائنات العلائقي لهذه المنظومة مع مراعاة معاني المصطلحات التالية:

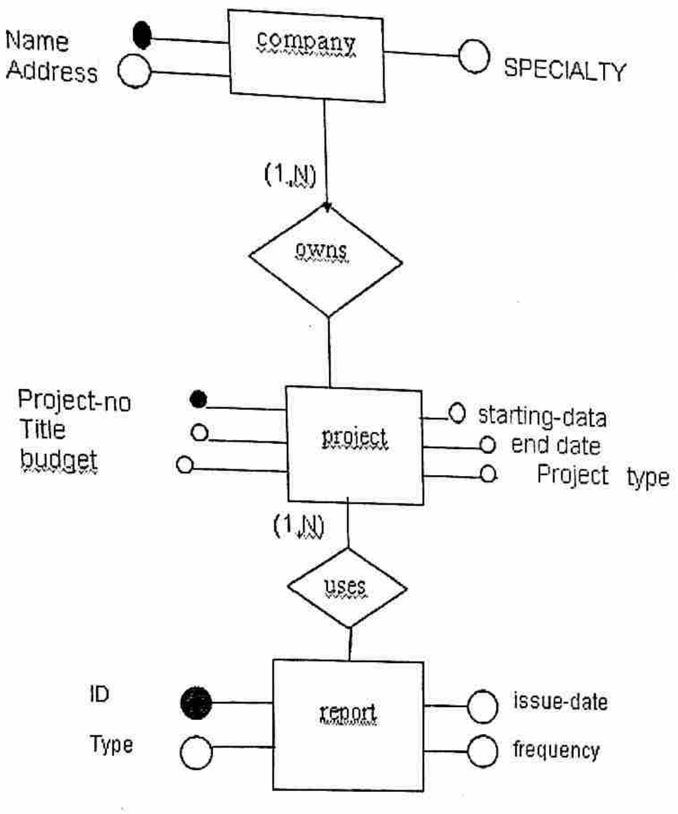
يملك **Owns**

تخصص **Specialty**

تاريخ إعداد التقرير **Issue Date** نوع التقرير **Type**

رمز التقرير **ID**

تكرار طباعة التقرير **Frequency** ميزانية المشروع **Budget**



قامـــوس البيانات Data Dictionary

تعريف: قاموس البيانات هو أداة تحليل وتصميم ويعتبر مخزن معلومات مركزي خاص ببيانات المنظومة.

وهذه البيانات تتمثل في عناصر البيانات أو الحقول.

ويمكن تعريف قاموس البيانات أيضا بأنه قائمة منظمة تحتوى على كل عناصر البيانات الخاصة بالمنظومة ليتم استخدامها من قبل المبرمج ومحلل النظم والمستخدم لتكون أداة تفاهم على نوع وخصائص بيانات المدخلات والمخرجات لأنها الأساس لتصميم قاعدة البيانات.

ويجب أيضا تحديد وتفصيل خصائص كل عنصر بيانات والتي من أهمها:

1. اسم الحقل (عنصر البيانات)
2. الاسم المختصر لعنصر البيانات
3. وصف نصي لعنصر البيانات
4. نوع عنصر البيانات
5. طول عنصر البيانات
6. نطاق ومدى قيم عنصر البيانات
7. القيمة الافتراضية لعنصر البيانات
8. مكان تخزين عنصر البيانات
9. موقع استخدام عنصر البيانات

نكتفي بعرض مثال مبسط لبعض هذه الخصائص :

1. اسم العنصر **Item-no**
2. مختصر العنصر **I-no**
3. وصف الحقل يستخدم كمفتاح رئيسي للوصول إلى سجل معين
4. نوع الحقل **Integer**
5. طول الحقل 4
6. نطاق الحقل 1 الى 9999

ويمكن تجهيز قاموس البيانات على شكل جدول أو تعبئة هدا القاموس التابع لنظم إدارة قواعد البيانات مثل **Access** أو **SQL** أو **ORACLE**.وهذا مثال لجدول قاموس بيانات لمنظومة مبيعات والجدول يوضح وصف لعناصر البيانات الخاصة بالكائن: زبون كما يلي:

جــدول زبـــون

اسم عنصر البيانات	اسم الحقل	النوع	الطول	النطاق	هل الحقل مفتاحي؟
رقم الزبون	CNO	INT	6	999999-1	نعم
أسم الزبون	CNAME	STRING	30		لا
العنوان	ADDER	STRING	30		لا
رقم الهاتف	PHONE	INT	8		لا
الرصيد	BALANCE	Real	10		لا

8.6 التصميم الإجرائي Procedural Design

يتم تجهيز التصميم الإجرائي **Procedural design** عادة بعد تجهيز التصميم المعماري وتصميم واجهة المستخدم .
ويتم تجهيز مواصفات الإجراء لكل جزء برمجي تمّ ذكره وتعريفه في التصميم المعماري بشيء من التفصيل.
وتستخدم أدوات لتمثيل هذه المواصفات و الخوارزميات باستخدام إمّا أدوات نصية (مثل شبه الشفرة **Pseudocode**) أو باستخدام المخططات مثل مخطط الانسياب الهيكلي **Structured flow chart**.

ويجب ملاحظة أن أي تطبيق يمكن أن يتم تصميمه إجرائيا باستخدام أحد التركيبات الهيكلية المتعارف عليها في البرمجة الهيكلية.

وهذه التركيبات هي : التركيبة التسلسلية **Sequence** والتركيبة الشرطية **Conditional** والتركيبة التكرارية أو ما يعرف بتركيبة الحلقة أو الدوران **Repetition**.
وفي البنود القادمة سنشرح بعض هذه الأدوات.

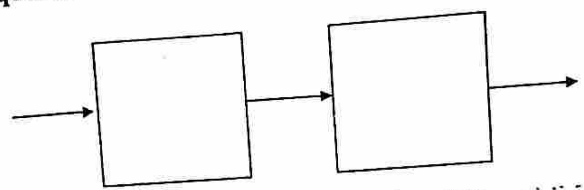
المخطط الانسيابي الهيكلي Structured Flow Chart

بالرغم من أن مفهوم استخدام الصورة بصفة عامة أفضل من النص لتوضيح أو تصميم أي شيء، إلاّ أن هذا المفهوم لايؤخذ على عمومه

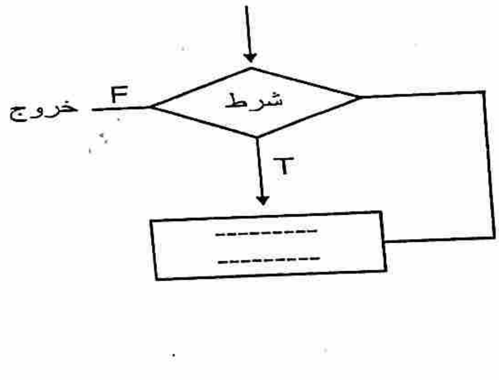
خصوصا في مجال البرمجة وإعداد المنظومات ، و خصوصا إذا كان هذا الرسم أو المخطط لا يفي بالمتطلبات أو يقود إلى تصميم غير جيد ، الأمر الذي سيقود بدوره إلى منظومة متدنية الكفاءة.
قد كان المخطط الانسيابي **Flowchart** في السابق الأداة الأكثر شهرة بين المبرمجين في الستينيات وحتى السبعينيات لتمثيل التصميم الإجرائي، ولكن نظرا لتعدد المتطلبات وكبر حجم المنظومات لم يستطع المخطط الإنسيابي أن يجاري هذا التغيير وبالتالي قادنا إلى برامج ومنظومات صعبة الفهم والصيانة وأصبحت الشفرة المنبثقة من المخطط الانسيابي تسمى شفرة سباقيتي **Spaghetti Code** (تشبيها لها بالأكلة الإيطالية المعروفة بتشابكها) وهي شفرة يكثر فيها استعمال جملة **Go to** وهي الجملة التي قد يؤدي استعمالها إلى صعوبة في وضوح وتتبع البرنامج .
إن المخطط الانسيابي القديم (**Flowchart**) يمكن أن يستخدم في تصميم البرامج الصغيرة أو شرح مفاهيم التصميم المبسط لتعليم الطلبة بعض مفاهيم تصميم البرامج، ولكنه بالتأكيد غير مناسب لتصميم المنظومات المعقدة والكبيرة.
والسؤال الذي يطرح نفسه هنا هو ماهو الحل البديل؟
إن الحل البديل لهذه المشكلة يكمن في تقسيم البرنامج الكبير أو المنظومة إلى أجزاء برمجية صغيرة ثمّ استخدام المخطط الانسيابي

لكل جزء، مستخدمين المخطط الانسيابي الهيكلي **Structured flowchart**، وهو عبارة عن أداة تصميم إجرائي (تطوير للمخطط الانسيابي القديم) مكون من أشكال هندسية تمثل التركيبات الثلاث (بنيات) الأساسية لأي برنامج كالآتي :

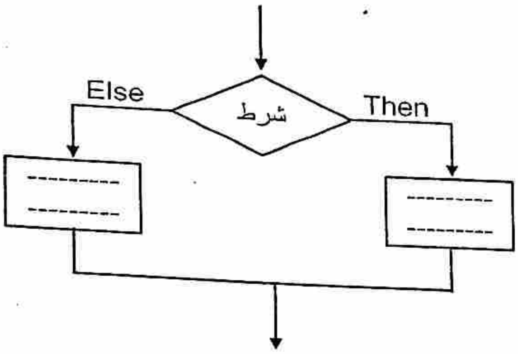
a) البنية التتابعية (أو التسلسلية) **Sequential Construct**



b) بنية نفذ- طالما **While-do Construct**



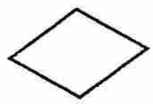
c) بنية إذا- وإلا
If –Then-Else Construct



لاحظ أن المستطيل يستعمل للمعالجة



و يستعمل الشكل التالي لاتخاذ القرار



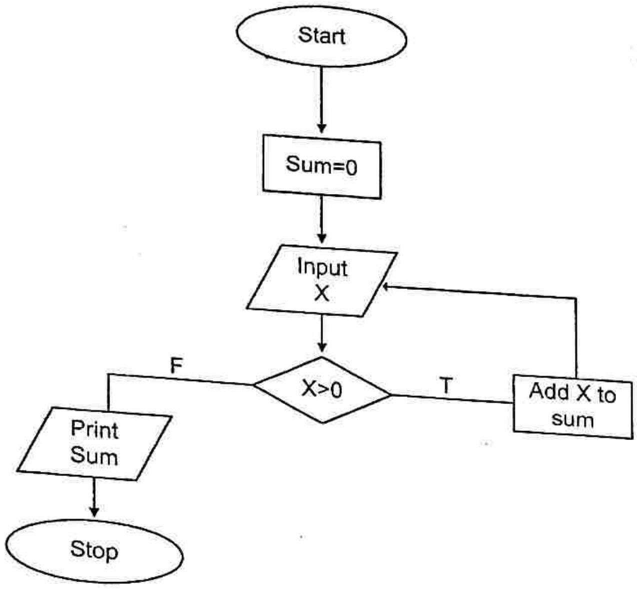
و الاختلاف بين المخطط الانسيابي القديم والمخطط الانسيابي الهيكلي هو أن المخططات الانسيابية الهيكلية تعتمد على استعمال مصطلحات البرمجة الهيكلية والتي منها **if-then-else** و **do-while**

مثـــال 1 :

ارسم مخطط انسيابي هيكلي لبرنامج يقرأ مجموعة أرقام موجبة ويطبع المجموع.

الحـــل :

في هذا المثال نوضح فكرة الحلقة ، حيث تقوم الحلقة بقراءة عدد وإضافته إلى المجموع طالما أن هذا العدد غير سالب ، وإلاّ فإن الحلقة تتوقف.



الشكل (**8.5.1**) مخطط انسيابي هيكلي لإيجاد المجموع

لاحظ أن تحويل هذا المخطط إلى شفرة برنامج سهلة جدا .

حاول تحويل هذا المخطط إلى برنامج (مثلا لغة بيسك المرئي).

مثــال 2

ارسم مخطط انسيابي هيكلي لمنظومة جباية فواتير الكهرباء حيث

تسعيرة الاستهلاك كما يلي :

- 2 درهم على الكيلووات الواحد إذا كان الاستهلاك أقل من 1000 كيلووات.
- 5 دراهم على الكيلووات الواحد إذا كان الاستهلاك 1000 كيلووات فما فوق.

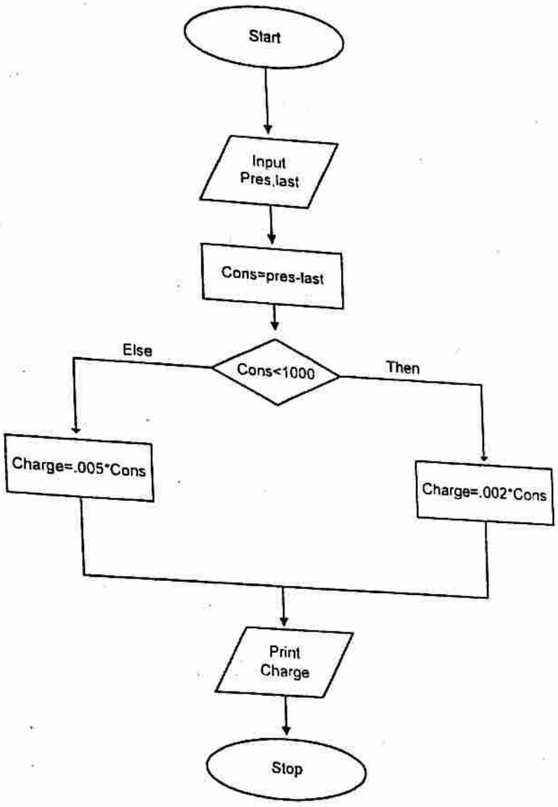
ارسم مخططا انسيابيا لطباعة فاتورة بحيث تبين الفاتورة القراءة السابقة والحالية للعداد والقيمة

الحــل:

نستخدم الرموز الآتية :

القراءة الحالية	**Pres**
القراءة السابقة	**Last**
الاستهلاك	**Cons**
القيمة المطلوبة (الفاتورة)	**Bill or Charge**

فيكون المخطط الانسيابي الهيكلي كما في الشكل (8.5.2) :



الشكل (**8.5.2**) مخطط انسيابي هيكلي لحساب تكاليف الكهرباء

8.7 شبـــه الشفرة Pseudocode

نستعرض هنا أداة أخرى لتمثيل التصميم الإجرائي بطريقة نصية بدلا من استخدام المخطط.

وتسمى هذه الأداة بشبه الشفرة **Pseudocode** لأنها تشبه إلى حد بعيد شفرة لغة برمجة مثل لغة بيسك المرئي أو لغة باسكال .

وتعتبر شبه الشفرة(أو شبيهة الشفرة) أداة تصميم إجرائي نصي تبين منطق البرنامج (خطوات الخوارزمية) بلغة دقيقة ومفهومة مستخدمة تركيبات البرمجة الهيكلية الثلاثة : التسلسلية والشرطية والتكرارية.

والمثال التالي يستخدم شبه الشفرة كأداة تصميم قبل ترجمتها إلى برنامج :

مثـــال 1 :

اكتب شبه الشفرة **Pseudo Code** لبرنامج يحسب فاتورة الكهرباء **(charge)** باستخدام القراءة السابقة للعداد **(last)** والقراءة الحالية **(Pres)** حسب الشروط الآتية:

إذا كانت كمية الاستهلاك بالكيلووات- ساعة **(cons)** اكبر من 1000 يكون سعر الكيلووات 50 درهم، وإذا كانت أقل أو يساوي 100 فيكون السعر 20 درهم.

الحـــل :

```
Read   pres, last
Cons = pres –last
If cons > 1000 then
    Charge= (50*cons) / 1000
Else
    Charge=(20*cons) / 1000
End if
Print pres, last, cons, charge
End
```

ملاحظة : نقسم الناتج على 1000 بغرض التحويل من درهم إلى دينار

مثـــال 2 :

اكتب شبه الشفرة **Pseudo Code** لحساب متوسط مجموع فواتير مبيعات محل بيع مواد بناء ليوم كامل . علما بأن عدد الفواتير مجهول ولكن توجد قيمة سالبة في النهاية كمؤشر لنهاية البيانات.

الحـــل : نقوم أولا بتحديد المتغيرات كالتالي :

قيمة مبيعات فاتورة	A
مجوع المبيعات ليوم كامل	S
متوسط مجموع الفواتير	AV
عدد الفواتير	N

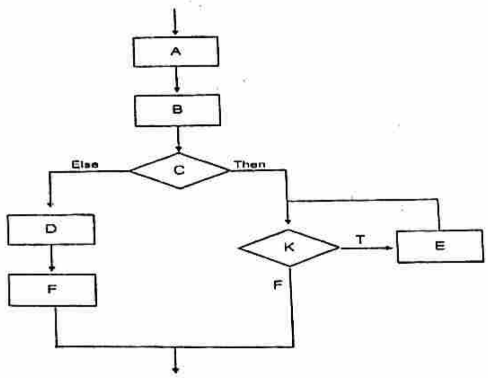
وبذلك تكون شبه الشفرة المطلوبة كما يلي :

```
S = 0
N = 0
S= 0
Input A
While   A > 0
     N = N+1
     S = S +  A
     Input A
  End  while
  AV = S /N
  Print AV
  End
```

مثـــال (3)

إن الغرض من هذا المثال هو توضيح التركيبات الثلاثة التسلسلية والشرطية والتكرارية (الحلقة) في مخطط انسيابي هيكلي واحد.

حيث يبين الشكل التالي مثالا لمخطط انسيابي هيكلي ، مع ملاحظة أن في هذا الشكل الرموز **A, B, D, F, E** تعبّر عن جمل مثل جملة التعيين أو الادخال أو الإخراج، ولكن الرمزين **C** و **K** هما عبارتان منطقيتان قيمتهما إما صواب أو خطأ.ولاحظ أيضا تحويل المخطط إلى شبه شفرة..



شبه الشفرة كما يلي:

```
Do   A
Do  B
IF   C   Then
        Do  D
          Do  F
ELSE
While (K)
          Do   E
Loop
End    IF
```

8.7 أسئلة للمراجعة

1. مامعنى المصطلح (تصميم)؟ اشرح المعنى وقارن بتصميم المباني.
2. ماهي نشاطات التصميم الرئيسية؟
3. عرّف "التصميم المعماري للنظم". اذكر مثالا لهذا التعريف.
4. عرّف "المخطط الهيكلي" واذكر رموزه.
5. ارسم مخططا هيكليا (ا) لمنظومة مخازن (ب) لمنظومة مبيعات مواد الكترونية..
6. عرّف مخطط ERD واذكر رموزه.
7. ارسم مخطط ERD لمنظومة مبيعات مواد منزلية.
8. لماذا يعتبر تصميم واجهة المستخدم مهما؟ وهل يجب أن يشارك المستخدم في هذا التصميم ولماذا؟
9. عرف ما يلي :
    - التصميم الهيكلي
    - شبيهة الشفرة
10. ارسم مخططا انسيابيا هيكليا لبرنامج يحسب متوسط أعمار 10 طلبة، ثم اكتب هذا البرنامج مستخدما شبه الشفرة.
11. ارسم مخططا انسيابيا هيكليا لبرنامج يحسب متوسط درجاتك حسب النظام المتبع في المؤسسة التعليمية التي تدرس بها، ثم اكتب هذا البرنامج مستخدما شبه الشفرة.

REFERENCES المراجع

1- Brooks F. P. (1975) . The Mythical Man Month. Reading MA: Addison-Wesly.

2- Constantine L. L. & Yourdine E. (1979) *Structured Design*. Englewood Cliffs NJ: Prentice-Hall.

3- Davis A. M. (1995) *201 Priciples of Software Developments*. McGraw Hill.

4- DeMarco T. (1978). *Structured Analysis and System Specification*. New York: yourdin Press.

5- Dijkstra E. W. (1976) A Displine of Programming. Englewood Cliffs NJ: Prentice-Hall.

6- Gane C. and Sarson T. (1975). *Structured System Analysis*. Englewood Cliffs NJ: Prentice-Hall.

7- Hamlet D. and Maybee j. (2001) The Engineering of Software. Addison-Wesly.

8- Pressman R. S. (1987) *Software Engineering – A Practitioner Approach*. 2nd edn. New York: McGraw Hill.

9- Shelly G. B. , Cashman T, J, Rosenblatt H. J. , (2001) Systems Analysis and Design, Boston: Course Technology.

10- Warnier J. D. (1977). Logical Construction of Programs. New York: Van Nostrand Reinhold.

انتهى الكتاب بفضل الله وتوفيق منه